



National Research
Council Canada

Conseil national
de recherches Canada

ERB-1010

Division of
Electrical Engineering

Division de
génie électrique

An Interactive Program on the IIS Display for an Adaptive System for Matching 3-D Range Data Objects

R. Kadamani and N.N. Abdelmalek

December 1987

88-04430/

Copyright 1987 by
National Research Council of Canada

Permission is granted to quote short
excerpts and to reproduce figures and tables
from this report, provided that the source of
such material is fully acknowledged.

Additional copies are available free of
charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Copyright 1987 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de
reproduire des figures ou tableaux du
présent rapport, à condition d'en identifier
clairement la source.

Des exemplaires supplémentaires gratuits
peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

AN INTERACTIVE PROGRAM ON THE IIS DISPLAY FOR
AN ADAPTIVE SYSTEM FOR
MATCHING 3-D RANGE DATA OBJECTS

R. Kadamani and W.N. Abdelmalek

ERB-1010
NRCC No. 28564

December 1987



TABLE OF CONTENTS

	Page
ABSTRACT	v
RESUME	v
1. INTRODUCTION	1
2. RUNNING THE PROGRAM	4
3. AN OVERVIEW OF THE SYSTEM	7
4. THE MAIN MENU	8
4.1 Perform Matching on Selected Objects	8
4.2 Display TESTED and/or MODEL Object Data	9
4.3 Display the Results in Curve Format	9
4.4 Display MODELS' Characteristics	10
4.5 Clear the IIS Display Screen CRT	12
4.6 Change the Colour Intensity Values	13
4.7 Colour the IIS Screen	13
4.8 Save Display	14
4.9 Restoring Last Saved Display	14
4.10 End	14
ACKNOWLEDGEMENTS	14
REFERENCES	14
APPENDIX A DOCUMENTATION OF THE SUBROUTINES	17
APPENDIX B PARAMETERS DEFINITIONS	29
APPENDIX C SOURCE LISTING OF THE DRIVER DRSHOW.FOR 39	

LIST OF FIGURES

1 The seven given real objects	2
2 Object 1 in four different orientations	4
3 The data set assign command	5
4 The link command	7
5 Flow chart of the main menu	7
6 Objects numbers	8
7 Flow chart of display menu	10
8 Four character curves of object no. 1 compared with a character curve of object no. 2 and a character curve of object no. 3. Shown also are four views of object no. 1	11
9 Four character curves of object no. 6 compared with a character curve of object no. 4 and a character curve of object no. 5. Shown also are four views of object no. 6	12
10 Flow chart for display of MODELS' characteristics	14



ABSTRACT

This report contains an interactive program on the IIS Display Monitor for an adaptive system for matching 3-D range data objects. The matching program constitutes an algorithm developed recently by N.N. Abdelmalek and the interactive program makes use of the IIS display system. Both the interactive program and the adaptive system are easily modified by the user. Extra sub-programs may be added to the system and existing ones may be easily bypassed.

RÉSUMÉ

Le présent rapport contient un programme interactif utilisable avec l'écran de visualisation IIS et destiné à un système adaptif permettant d'assortir des objets tridimensionnels à partir de données sur les distances. Le programme d'assortiment comporte un algorithme mis au point récemment par N.N. Abdelmalek et le programme interactif est utilisé avec l'écran de visualisation IIS. Le programme interactif et le système adaptif sont facilement modifiables par l'utilisateur. Des sous-programmes facultatifs peuvent être ajoutés au système et les sous-programmes existants peuvent être facilement court-circuités.



**AN INTERACTIVE PROGRAM ON THE IIS DISPLAY FOR
AN ADAPTIVE SYSTEM FOR
MATCHING 3-D RANGE DATA OBJECTS**

R. Kadamani and W.N. Abdelmalek

1. INTRODUCTION

An interactive program "DRSHOW.FOR" on the IIS screen display for matching 3-D range data objects is here described. The matching program constitutes an algorithm developed recently by the second author and is described in detail in ref. 1. The interactive program makes use of the IIS display system. Both the interactive program and the adaptive system are easily modified by the user. Extra sub-programs may be added and existing ones may be easily bypassed.

In the present program, we experiment with seven industrial-like objects and two simulated ones. One of the simulated objects is an ellipsoid and the other one a paraboloid of revolution. Figure 1 shows one view of each of the seven given real objects.

Each of objects 1, 2, and 8 is translated and rotated an arbitrary angle about the object z-axis four times. Each of the other objects is translated and rotated five times. Figure 2 shows object 1 in the four different orientations.

The z-coordinates of the 42 data sets are stored in the computer in matrix form. The (i, j) matrix element is the z-coordinate of the object at the point (i, j) or (x, y).

Object 1 has the four data sets BLOCK0A.YIN, BLOCK1A.YIN, BLOCK8A.YIN, and BLOCK9A.YIN. Object 2 has the four data sets BLOCK10A.YIN, BLOCK15A.YIN, BLOCK16A.YIN, AND BLOCK17A.YIN, and so on. The 42 object data are assigned the logical unit numbers FOR015 to FOR047 for the real objects and FOR051 to FOR059 for the simulated objects. See Fig. 3.

The data sets BLOCK0A.YIN, BLOCK10A.YIN, BLOCK111.YIN, BLOCK116.YIN, BLOCK1111.YIN, BLOCK116.YIN, BLOCK1121.YIN, SIMULFIG2.DAT, and SIMULFIG6.DAT are chosen for the nine objects as "Reference" or "Model" data. See Fig. 3 again.

The function of the present program is to match each of the remaining 33 data set to its model set. For example, each of the data sets BLOCK1A.YIN, BLOCK8A.YIN, and BLOCK9A.YIN should be matched to object 1, or each of these three data sets would be those of object 1.



Figure 1. The seven given real objects

For each of the "Model" data, a set of 15 invariant matching parameters are calculated and stored in the matching library. As soon as any one of the 33 mentioned data sets is read by the computer, the set of 15 matching parameters is calculated and compared to the nine sets of 15 matching

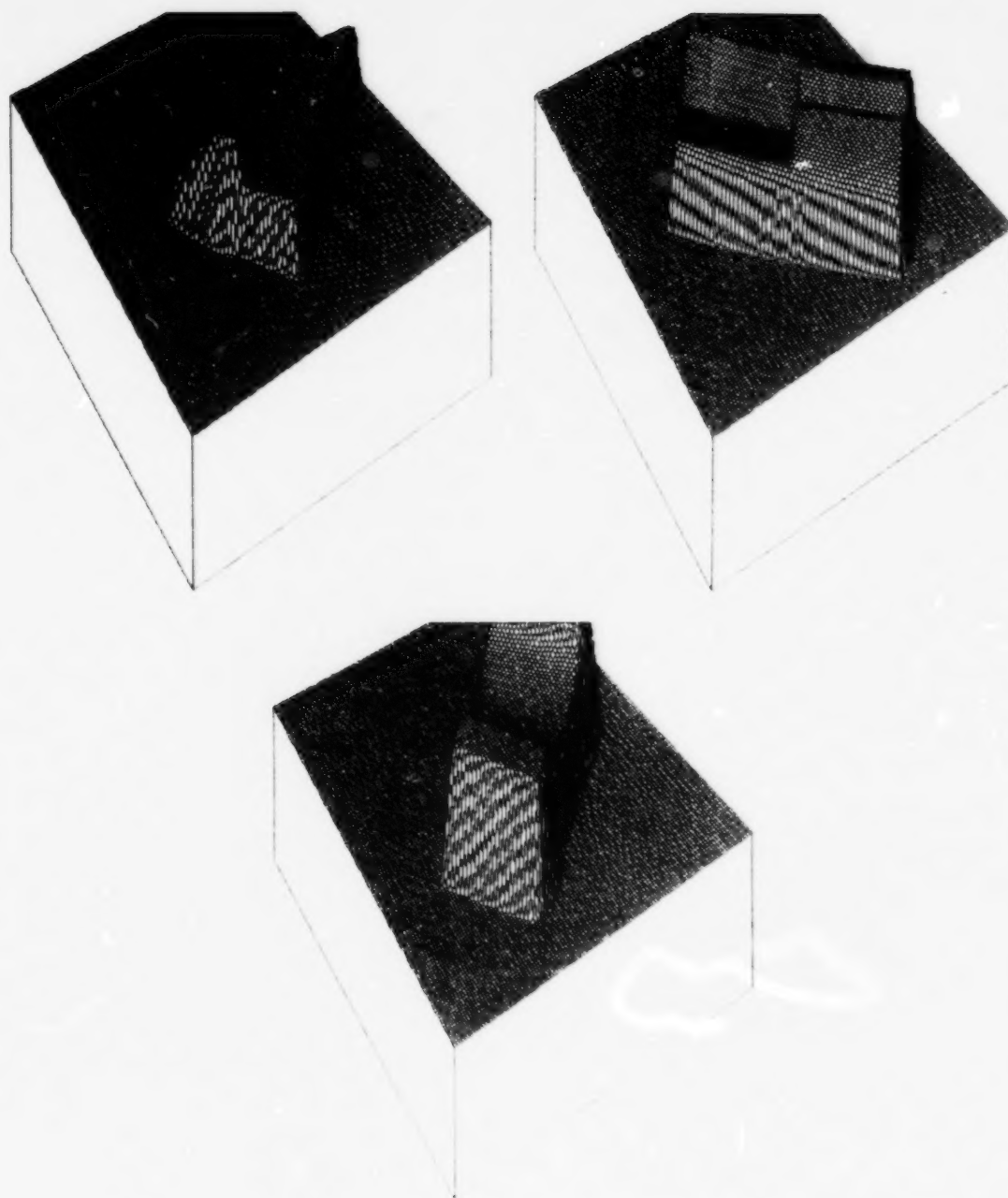


Figure 1. (Concluded)

parameters of the models. The program successfully matches the read data to its model. The present program operates in an interactive mode on the IIS screen display.



Figure 2. Object 1 in four different orientations

2. RUNNING THE PROGRAM

Program DRSHOW.FOR is the "Driver" for running this system. It is written in FORTRAN 77 and is implemented on the VAX 11/780 and the IIS Image Display Monitor. As usual, to utilize "DRSHOW.FOR," the following steps are taken.

```

C
C-----< PROCEDURE ASSIGN.COM >-----
C
$ SET NOVERIFY
$ ASS DUAO:[IIS.TASK] S500_TASK
$ ASS DUAO:[IIS.TASK] T
$
$ ASS [ABDELMale.PICTURE]BLOCK0A.YIN FOR015
$ ASS [ABDELMale.PICTURE]BLOCK1A.YIN FOR016
$ ASS [ABDELMale.PICTURE]BLOCK8A.YIN FOR017
$ ASS [ABDELMale.PICTURE]BLOCK9A.YIN FOR018
$
$ ASS [ABDELMale.PICTURE]BLOCK10A.YIN FOR019
$ ASS [ABDELMale.PICTURE]BLOCK15A.YIN FOR020
$ ASS [ABDELMale.PICTURE]BLOCK16A.YIN FOR021
$ ASS [ABDELMale.PICTURE]BLOCK17A.YIN FOR022
$
$ ASS [ABDELMale.PICTURE]BLOCKII1.YIN FOR023
$ ASS [ABDELMale.PICTURE]BLOCKII2.YIN FOR024
$ ASS [ABDELMale.PICTURE]BLOCKII3.YIN FOR025
$ ASS [ABDELMale.PICTURE]BLOCKII4.YIN FOR026
$ ASS [ABDELMale.PICTURE]BLOCKII5.YIN FOR027
$
$ ASS [ABDELMale.PICTURE]BLOCKII6.YIN FOR028
$ ASS [ABDELMale.PICTURE]BLOCKII7.YIN FOR029
$ ASS [ABDELMale.PICTURE]BLOCKII8.YIN FOR030
$ ASS [ABDELMale.PICTURE]BLOCKII9.YIN FOR031
$ ASS [ABDELMale.PICTURE]BLOCKII10.YIN FOR032
$
$ ASS [ABDELMale.PICTURE]BLOCKII11.YIN FOR033
$ ASS [ABDELMale.PICTURE]BLOCKII12.YIN FOR034
$ ASS [ABDELMale.PICTURE]BLOCKII13.YIN FOR035
$ ASS [ABDELMale.PICTURE]BLOCKII14.YIN FOR036
$ ASS [ABDELMale.PICTURE]BLOCKII15.YIN FOR037
$
$ ASS [ABDELMale.PICTURE]BLOCKII16.YIN FOR038
$ ASS [ABDELMale.PICTURE]BLOCKII17.YIN FOR039
$ ASS [ABDELMale.PICTURE]BLOCKII18.YIN FOR040
$ ASS [ABDELMale.PICTURE]BLOCKII19.YIN FOR041
$ ASS [ABDELMale.PICTURE]BLOCKII20.YIN FOR042
$

```

Figure 3. The data sets assign command

```

$ ASS [ABDELMale.PICTURE]BLOCKII21.YIN FOR043
$ ASS [ABDELMale.PICTURE]BLOCKII22.YIN FOR044
$ ASS [ABDELMale.PICTURE]BLOCKII23.YIN FOR045
$ ASS [ABDELMale.PICTURE]BLOCKII24.YIN FOR046
$ ASS [ABDELMale.PICTURE]BLOCKII25.YIN FOR047
$
$ ASS [ABDELMale.VISION]SIMULFIG1.DAT FOR051
$ ASS [ABDELMale.VISION]SIMULFIG2.DAT FOR052
$ ASS [ABDELMale.VISION]SIMULFIG3.DAT FOR053
$ ASS [ABDELMale.VISION]SIMULFIG4.DAT FOR054
$
$ ASS [ABDELMale.VISION]SIMULFIG5.DAT FOR055
$ ASS [ABDELMale.VISION]SIMULFIG6.DAT FOR056
$ ASS [ABDELMale.VISION]SIMULFIG7.DAT FOR057
$ ASS [ABDELMale.VISION]SIMULFIG8.DAT FOR058
$ ASS [ABDELMale.VISION]SIMULFIG9.DAT FOR059
$
$ ASS [ABDELMale.VISION]OBJECTL.DAT FOR070
$
$ ASS [ABDELMale.VISION]BLOCK0AL.DAT FOR071
$ ASS [ABDELMale.VISION]BLOCK10AL.DAT FOR072
$ ASS [ABDELMale.VISION]BLOCKII1L.DAT FOR073
$ ASS [ABDELMale.VISION]BLOCKII6L.DAT FOR074
$ ASS [ABDELMale.VISION]BLOCKII11L.DAT FOR075
$ ASS [ABDELMale.VISION]BLOCKII16L.DAT FOR076
$ ASS [ABDELMale.VISION]BLOCKII21L.DAT FOR077
$
$ ASS [ABDELMale.VISION]SIMULFIG2L.DAT FOR078
$ ASS [ABDELMale.VISION]SIMULFIG6L.DAT FOR079
$
$ ASS[FOR003.DAT FOR004
$ ASS TT FOR003
$ ASS TT FOR005
$
$ SET NOVERIFY

```

Figure 3. (Concluded)

1. The object data sets are assigned to their logical unit numbers. See Fig. 3, \$ @ASSIGN
2. The main program DRSHOW.FOR is compiled, \$ FOR DRSHOW

3. The main program is linked to the Object Library. See Fig. 4,
\$ @LKDRSHOW and
4. The main program is run, \$ RUN DRSHOW

```

C-----< PROCEDURE LKDRSHOW.COM >-----

$ SET VERIFY
$ LINK DRSHOW,-
DUAO:[IIS.TASK]S465/LIB,-
DRB1:[ABDELMAL.VISION]VISION/LIB,-
DRB1:[ABDELMAL.PICTURE]PICTURE/LIB
$ SET NOVERIFY

```

Figure 4. The link command

3. AN OVERVIEW OF THE SYSTEM

A simplified flow-chart for the system is shown in Fig. 5. Basically when the Driver "DRSHOW" runs, first of all, a "Main Menu" appears on the

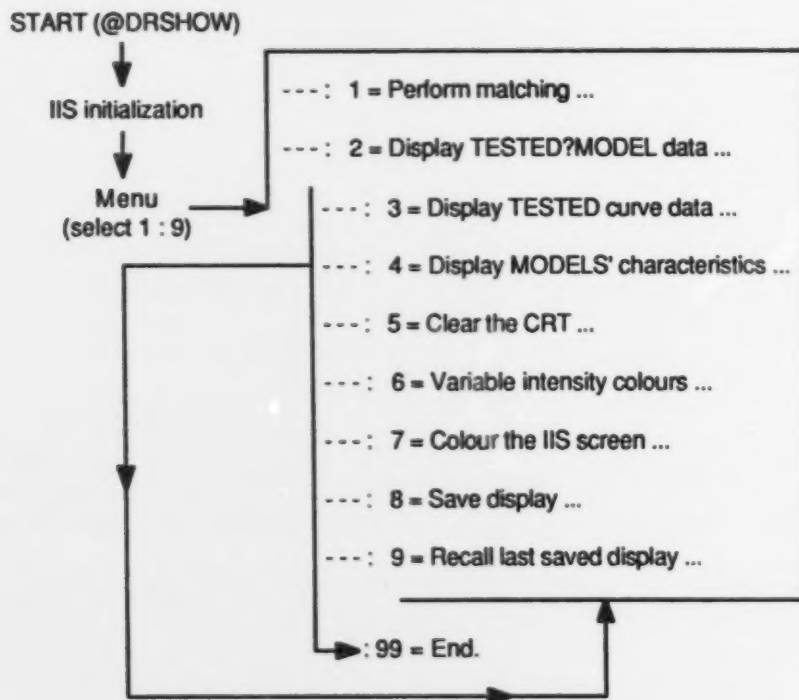


Figure 5. Flow chart of the main menu

terminal screen. The system then responds to the choices of the user, both for selecting from the menu and for the parameters needed to run the selected option. Upon completion of execution of the selected option the system redisplay the main menu.

4. THE MAIN MENU

As the flow chart of Fig. 5 shows, the menu has nine options. To start with, the user obviously has to select option 1 for the "Matching" Operation. Next, the user may select any one of the nine menu options. The items of the menu are shown here and are explained one by one.

1. Perform matching on selected objects data
2. Display TESTED and/or MODEL image data
3. Display TESTED data results in "curve format"
4. DEMO (display MODEL'S characteristics)
5. Clear CRT
6. Change colour intensities
7. Colour the IIS screen
8. Save display
9. Restore last saved display
99. END

4.1 Perform Matching on Selected Objects

Here, the user has the following options:

1. The user selects the object's number KOBJ, [1:42] (Fig. 6), for which the invariant matching characteristics are calculated and compared to the nine sets of invariant matching characteristics of the nine models.

```
C-----< Select Tested Object Number 'KOBJ' >-----
C
```

```
KOBJ= 1:4  -> Object 1
KOBJ= 5:8  -> Object 2
KOBJ= 9:13 -> Object 3
KOBJ= 14:18 -> Object 4
KOBJ= 19:23 -> Object 5
KOBJ= 24:28 -> Object 6
KOBJ= 29:33 -> Object 7
KOBJ= 34:37 -> Object 8
KOBJ= 38:42 -> Object 9
```

Figure 6. Objects numbers

2. A Gaussian noise of zero mean and standard deviation "SIGMANOIS" may be added to the selected data. The range of "SIGMANOIS" is [0.0:4.0].
3. The selected object data, with or without the added Gaussian noise, may or may not be smoothed.
4. If smoothing of the object data is required, a smoothing window size of 3×3 , 5×5 , or 7×7 is selected.
5. The user has the choice of creating a "FOR003.DAT" file and printing it later. This file is a copy of the calculated results. If yes, set IFOR003 flag = 1.
6. The user may select to display the object on the terminal screen (not the IIS terminal). The display is in character form, such that each of the object elements takes one of the characters 0123456789 in an ascending order.
7. The user may also wish to display the contour of the base silhouette on the terminal screen.

At this stage, the matching operation is done and the matching results are ready to be displayed on the IIS monitor. We mention here that, on the average, reading a data set takes about 10 s and running the matching algorithm takes about 7 s of CPU time.

4.2 Display TESTED and/or MODEL Object Data

This menu selection will display on the terminal screen the following sub-menu and offers the user one of three choices.

1. To display the MODEL object of the tested object on the IIS
2. To display the tested object on the IIS monitor, or
3. Exit (to the main menu).

If the selection is 1 or 2, the display may be magnified on the IIS screen up to eight times its size. Also the user displays the object from a starting column and a starting row. See Fig. 7.

4.3 Display the Results in "Curve Format"

The 15 invariant matching parameters obtained at the end of section 4.2 above are scaled and may be displayed on the IIS in the form of a

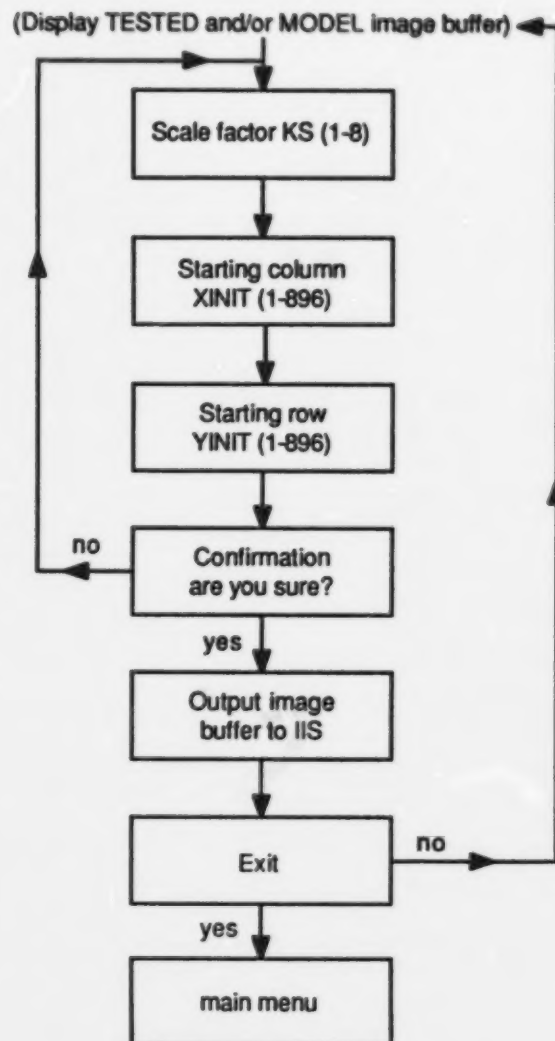


Figure 7. Flow chart of display menu

connected curve. That is to simplify comparing this result with the already stored models results, which are also scaled and stored in the form of curves. See Figs. 8 and 9.

4.4 Display of MODELS' characteristics

This option is needed when the user wishes to plot the 15 invariant matching parameters in curve format for one, two or more ..., or for all of the nine models.

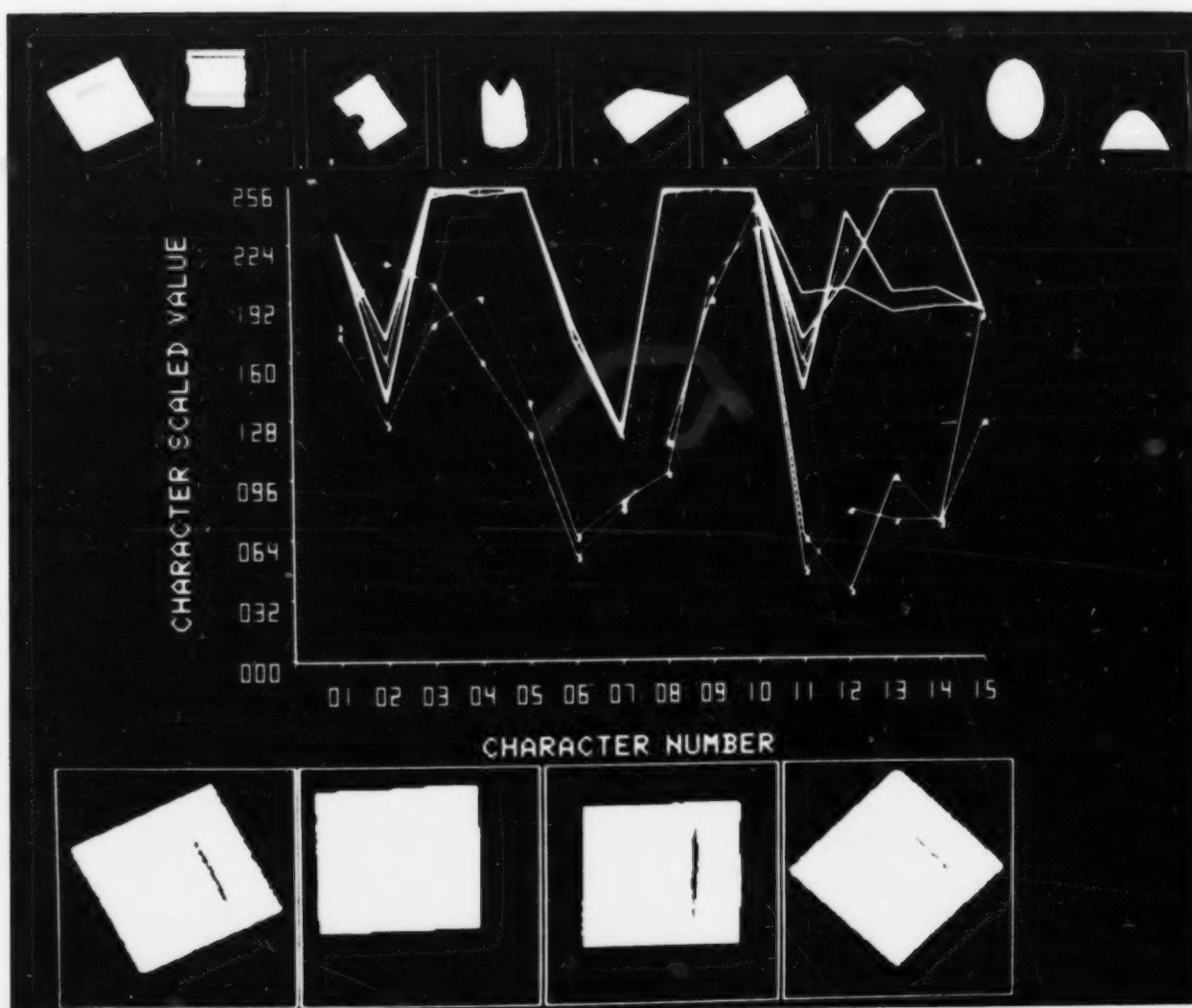


Figure 8. Four character curves of object no. 1 compared with a character curve of object no. 2 and a character curve of object no. 3. Shown also are the four views of object no. 1

This option with that of Section 4.3 helps one to decide quickly, by simply looking at the plots, to which of the nine models the tested object belongs.

The subroutine which utilizes this option is called "DEMO.FOR". Figure 10 is a flow chart of the "DEMO" program.

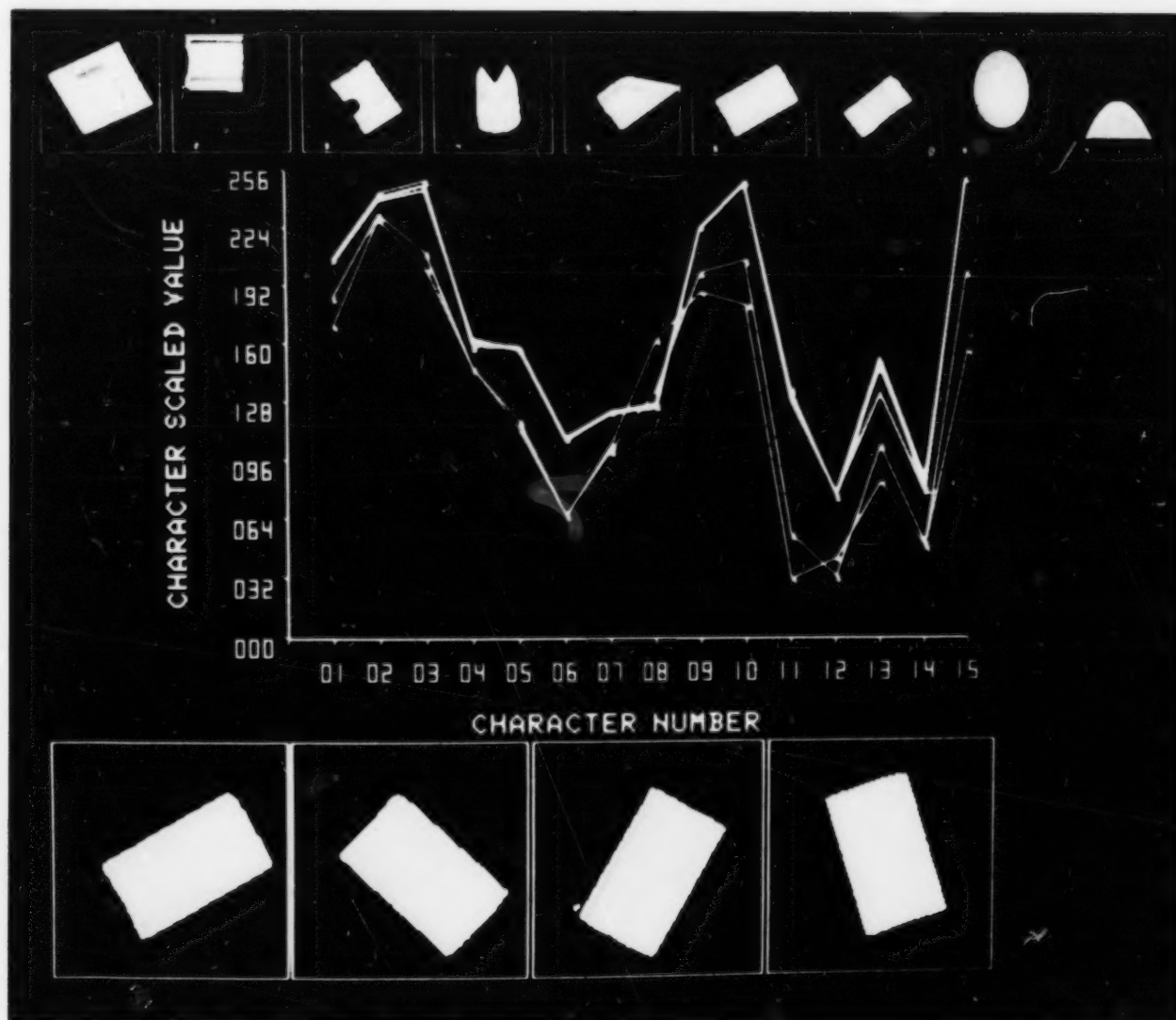


Figure 9. Four character curves of object no. 6 compared with a character curve of object no. 4 and a character curve of object no. 5

4.5 Clear the IIS Display Screen CRT

This option clears the IIS display, in order to display new items without crowding the IIS monitor screen.

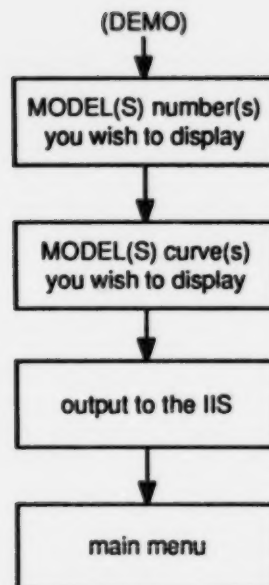


Figure 10. Flow chart for display of MODELS' characteristics

4.6 Change the Colour Intensity Values

This option allows the user to change the intensity values for each of the RED, GREEN, and BLUE look-up tables up to nine times.

4.7 Colour the IIS Screen

This option in the main menu is for colour control of the IIS display facility, the following selections are available:

1. SAW-TOOTH ramp (user will supply the wavelength). The user may select the period of the SAW-TOOTH wave-form which is actually the ramp to be loaded in the LOOK-UP tables. The user may also select the maximum height within the period.
2. LINEAR ramp (go back to image standard values). With this selection the LOOK-UP tables will be loaded with a linear ramp which means that the display will be shown in a GRAY-LEVEL scale.
3. CONTROLLED ramp (user will supply end points). If this option is selected the user can load each of the LOOK-UP tables with a selected ramp; the starting and ending points are user selectable. Also the

user has the option to turn off any one of the LOOK-UP tables or any combination of them, i.e., if for example it is desired to turn off the RED colour, the user inputs zero values at the prompt for starting and ending points for the RED colour ramp. It is also possible to get the negative of the image by loading a negative ramp, i.e., the ramp will start from a maximum and end by a minimum.

4.8 Save Display

Upon selection of this option, whatever is displayed in image memory 0 will be saved in image memory 1. This option is useful if the user wishes to recall the last saved display, especially after clearing the CRT of the IIS. Note that this option uses the fast Arithmetic Logic Unit (ALU) of the IIS. Thus the display is saved in a fraction of a second.

4.9 Restoring Last Saved Display

This option is invoked if the user wishes to replace the display image memory 0 with the display of image memory 1 which was originally saved with the Save option described above. Also this option uses the (ALU) of the IIS.

4.10 End

This is usually the user's last option. Upon selecting this option the interactive part of the DRSHOW.FOR will be terminated and the control will be transferred to the operating system.

ACKNOWLEDGEMENTS

The authors had stimulating discussions with Nestor Burtnyk, Shadia El-Gazzar, and Marc Rioux. Kenneth Evans, Arthur Binch, and Michael Duggan helped with the IIS display monitor, the VAX system and the camera. We are deeply thankful to all of them. A special thanks is to Luc Cournoyer for providing us with the range data. We also wish to thank Evelyn Kidd and Theresa Tierney, the staff of the Editorial Office of the Division of Electrical Engineering of the NRC, for their assistance in the publication of this and our other reports.

REFERENCES

1. N.N. Abdelmalek. A computer vision system for matching 3-D range data objects, NRC/ERB-995, National Research Council of Canada, Ottawa, Ont. September 1987.

2. M.-K. Hu. Visual pattern recognition by moment invariants, IRE Trans. Inf. Theory, IT-8: 179-187; 1962.
3. R.Y. Wong and E.L. Hall. Scene matching with invariant moments. Comput. Vision, Graphics, Image Process. 8: 16-24; 1978.
4. E.L. Hall. Computer Image Processing and Recognition. Academic Press, 1979.
5. S.A. Dudani, K.F. Breeding, and R.B. McGee. Aircraft identification by moment invariants. IEEE Trans. Comput. C-26(1): 39-45; 1977.
6. S. Maitra. Moment invariants. Proc. IEEE, 67: 697-699; 1979.
7. E. Kreyszig. Introduction to Differential Geometry and Riemannian Geometry. University of Toronto Press, Toronto, Ont. 1968.
8. P.J. Besl and R. Jain. Invariant surface characteristics for 3D object recognition in range images. Comput. Vision, Graphics, Image Process. 33: 33-80; 1986.
9. W.I. Grosky and R. Jain. Pyramid-based approach to segmentation applied to region matching. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8: 639-650, 1986.
10. Y.F. Wang, M.J. Magee, and J.K. Aggarwal. Matching three-dimensional objects using silhouettes. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6: 513-518; 1984.
11. F.L. Alt. Digital pattern recognition by moments. ACM J. 9: 240-258; 1962.
12. M. Rioux. Laser range finder on synchronized scanners. Appl. Opt. 23: 3837-3844; 1984.
13. F. Bumbaca, F. Blais, and M. Rioux. Real-time correction of three-dimensional nonlinearities for a laser rangefinder. Opt. Eng. 25(4): 561-565; 1986.
14. F. R. Livingstone and M. Rioux. Optical techniques for industrial inspection, Proc. SPIE - The International Society for Optical Engineering, 665: 188-194; 1986.
15. F.B. Seely, N.E. Ensign, and P.G. Jones. Analytical Mechanics for Engineers. 5th Edition. John Wiley, New York. 1958.

APPENDIX A. DOCUMENTATION OF THE SUBROUTINES

A1. The IIS Initialization

For handling IIS display facilities, the IIS first of all should be initialized. This is done by a numbers of calls to the specific subroutines. These routines are called by the Driver 'DRSHOW.FOR'. For example calling FCBSET will initialize the Function Communication Bloc, OPM75 will initialize Model 65 options, GTM70 will acquire and initialize the display and see if the display is available. The call for MOD640 will setup the display to one of the available modes, the call for SETUP will enable channels associated with a given image and initialize various registers, and so on. For more detail, refer to DRSHOW.FOR listing below.

A2. Documentation of the Subroutines

There are two major sets of subroutines, the first set deals with handling of the IIS display. The second set deals with the matching process and finding the best match for the tested object.

A2.1 The FIRST set of subroutines

A2.1-1 DEMO

```

C-----
C          SUBROUTINE DEMO ( FCB, JSW, ISCALEDAT, ICHARCTR )
C-----
C      This routine is invoked if the user wishes to plot in graphic form
C the results after the matching process has been completed. The plot consists of displaying the nine models in [128*128] image form on top of the screen. Then the routine will draw the y-(scale) axis and the x-axis which corresponds to the parameters numbers, with their corresponding labels. The routine then sketches one or more of the nine curves of the invariant parameters which correspond to the nine models.
C
C Parameters declaration:
C
C FCB      : Integer Array (IA) Function Communications Block. This array
C           is used for all communication between various subsystems of
C           the System 575.
C JSW = 0  : For displaying tested curve data and exit to main menu.
C JSW = 1  : For displaying the models, axes, labels, curves, etc.... and
C           exit to main menu.
C ISCALEDAT: Integer array.          It contains the results of vector SHOW,
C           scaled to [0:255].
C ICHARCTR : An integer; the number of the invariant parameters.

```

C
 C Note: Subroutine DEMO calls the subroutines V_LINE, H_LINE, V_MARK,
 C H_MARK, PCHAR, PCHARH; (which are found in DEMO), CH0, CH1, CH2,
 C CH3, CH4, CH5, CH6, CH7, CH8, CH9; (which are found in CDISPLAY),
 C CHC, CHA, CHR, CHT, CHE, CHS, CHL, CHD, CHV, CHN, CHU, CHM, CHB;
 C (found DEMO).

C

C-----
 C SUBROUTINE V_LINE(FCB,NLINE,YSTART,YEND,XINIT)
 C-----

C This routine is used to draw a solid vertical line. The starting/
 C ending positions, thickness of the line and the position of the line are
 C selected by the user.

C Parameters declaration:

C

C

C FCB : Integer Array (IA), defined above.
 C YSTART : An integer; the starting position along Y_AXIS (vertical
 C position).
 C YEND : An integer; the position along Y_AXIS.
 C NLINE : An integer; the thickness of the line.
 C XINIT : The line position (column = XINIT).

C

C-----
 C SUBROUTINE H_LINE(FCB,NLINE,XSTART,XEND,YINIT)
 C-----

C This routine is used to draw a solid horizontal line. The starting/
 C ending positions, thickness of the line and the position of the line are
 C selected by the user.

C Parameters declaration:

C

C XSTART, XEND, NLINE, YINIT : Similar parameters to those of Subroutine
 C V_LINE, except for a horizontal line.

C

C-----
 C SUBROUTINE V_MARK(FCB,MSTEP,MSIZE,YSTART,YEND,XINIT)
 C-----

C This routine is used to divide the parameter scaled value axis into
 C intervals of equal length.

C

C Parameters declaration:

C

C YSTART : An integer; the starting position along Y_AXIS (vertical
 C position).

C YEND : An integer; the position along Y_AXIS.
 C NLINE : An integer; the thickness of the line.
 C XINIT : The line position (column = XINIT).
 C MSIZE : The thickness of the mark.
 C MSTEP : The distance between two marks.

C

C-----

C SUBROUTINE H_MARK(FCB,MSTEP,MSIZE,XSTART,XEND,YINIT)

C-----

C This routine is used to divide the parameter number axis into
 C fifteen intervals of equal length.

C

C Parameters declaration:

C

C XSTART, XEND, YINIT, MSIZE, MSTEP: Similar parameters to those of
 C Subroutine V_MARK, except that they for the horizontal
 C direction.

C-----

A2.1-2 CDISPLAY

C-----

C SUBROUTINE CDISPLAY (Ibuff, Kobj, IDIS)

C-----

C This routine is used to write the character fond in Ibuff at the
 C lower left corner of the image buffer.

C Parameters declaration:

C

C Parameters declaration:

C

C-----

C Kobj : An integer, the object number, chosen for the test the matching
 C process; [Kobj = 1:42].

C IDIS : an integer number

C = 1 , the user wants model data to be displayed

C = 2 , the user wants object data to be displayed

C = 3 , the user wants to exit to the main menu.

C Ibuff : A [128*128] integer matrix.

C

C Note : Routine CDISPLAY calls the following subroutines

C

C-----

```

C                               SUBROUTINE CHC(IBUFF)
C-----
C      This routine is used to generate the font for letter 'C'.
C-----
C      SUBROUTINES CHH(IBUFF), CHA(IBUFF), CHR(IBUFF), CHT(IBUFF),
C CHE(IBUFF), CHS(IBUFF), CHL(IBUFF), CHD(IBUFF) CHV(IBUFF), CHU(IBUFF),
C CHN(IBUFF), CHM(IBUFF), and CHB(IBUFF) are subroutines used to generate
C the fonts H, A, R, T, E, S, L, D, V, U, N and M respectively.
C-----

```

A2.1-3 LINLUT

```

C-----
C                               SUBROUTINE LINLUT ( FCB,PI,IBUFF )
C-----
C      This routine is used to change the intensity values for each of the
C RED, GREEN and the BLUE look-up tables up to nine times.
C
C Parameters declaration:
C
C FCB      : Integer Array (IA), defined above.
C PI       : a real number  $PI = 4.0 * ATAN(1.)$ 
C IBUFF    : A [128*128] integer matrix.
C-----
C

```

A2.1-4 IISLINLUT

```

C-----
C                               SUBROUTINE IISLINLUT ( FCB )
C-----
C      This subroutine displays on the screen a sub-menu of 3 options:
C
C      1: SAW-TOOTH ramp (user will supply the wavelength). The user may
C select the period of the SAW-TOOTH wave-form to be loaded in the LOOK-UP
C tables. The user may also select the maximum height within the period.
C
C      2: LINEAR ramp (go back to image standard values). With this
C selection the LOOK-UP tables are loaded with a linear ramp which means
C that the display will be shown in a GRAY-LEVEL scale.
C
C      3: CONTROLLED ramp (user will supply end points). If this option
C is selected the user may load each of the LOOK UP tables with a selected

```

C ramp. The starting and ending points are the user selectable. Also the
 C user has the option to turn off any one of the LOOK-UP tables or any
 C combination of them. That is, if for example it is desired to turn off
 C the RED colour, then user inputs zero values at the prompt for starting
 C and ending points for the RED colour ramp. It is also possible to get
 C the negative of the image by loading a negative ramp, i.e., the ramp
 C will start with a maximum and ends with a minimum value.

C

C Parameters declaration:

C

C FCB : Integer Array (IA), defined above.

C

C-----

A2.2 The SECOND Set of Subroutines

A2.2-1

C-----

C

*****SUBROUTINE READIMAGE does the following.

C

C It reads a given integer file of dimensions 128x128 and unit number
 C 'IUNIT' and outputs it in the form of a real 128x128 matrix 'PICTUR'. It
 C also obtains the largest element of PICTUR(I,J).

C

C If desired by the user, a GAUSSIAN noise of zero mean and standard
 C deviation sigmanois may be added to the nonzero elements of the input
 C file. That is to the object itself not to the background.

C-----

A2.2-2

C-----

C

*****SUBROUTINE READIMAG2 does the following.

C

C It reads a given real*4 file of dimensions 256x256 and unit number
 C 'IUNIT' obtained by the range camera and reduces it to a real 128x128
 C matrix 'PICTUR' by taking every other pixel every other row. It also
 C obtains the largest elements of the read file.

C

C If desired by the user, a GAUSSIAN noise of zero mean and standard
 C deviation 'SIGMANOIS' may be added to the input nonzero elements of the
 C input file. That is to the object itself not to the background.

C-----

A2.2-3

```

C-----
C          *****SUBROUTINE BOUNDS does the following.
C      Given is a 3-d object in the form of a matrix 'SPICTUR' whose Z-
C      coordinates are the (I,J) values in the matrix.
C      This subroutine calculates: (1) the (I,J) coordinates of the corners
C      of the rectangle inside which the nonzero values of Z are enclosed. One
C      obtains IMIN, IMAX, JMIN AND JMAX. (2) the I and J-coordinates of the
C      nonzero Z-coordinates of the object in the form of a string of the X-
C      coordinates (FLOAT(I)-coordinates) and the corresponding strings of the
C      Y-(FLOAT(J)) and the Z-coordinates, and (3) the number of such nonzero
C      Z-coordinates. (4) It also creates a 'LOGICAL*1' file matrix 'MAPL' of
C      the input file stores under the unit number 'JUNIT' for display purposes
C      on a display screen.
C-----

```

A2.2-4

```

C-----
C          *****SUBROUTINE READSTRNG does the following.
C
C      This subroutine reads a given real string [X(1),Y(1),Z(1),
C      X(2),Y(2),Z(2), ... , X(LAST),Y(LAST),Z(LAST)]. It outputs 3 vectors,
C      for the X-, Y- AND Z- corresponding coordinates respectively. It also
C      calculates the largest values of the X, Y AND Z coordinates and the
C      smallest X and Y coordinates of the given image.
C      If desired by the user, a GAUSSIAN noise of zero mean and standard
C      deviation 'SIGMANOIS' may be added to the Z- values of the input file.
C-----

```

A2.2-5

```

C-----
C          *****SUBROUTINE OBJECT does the following.
C      This subroutine does the following.
C      (1) It creates simple symmetric geometrical objects of different
C      sizes; SPHERES, ELLIPSOIDES, PARABOLOIDS OF REVOLUTION and CONES to be
C      used for testing the invariant parameters of the object, such as the
C      curvature and the moment parameters. The created object is given by
C      matrix 'PICTUR' whose (I,J) element is the Z-coordinate of the object.
C      If desired by the user, a GAUSSIAN noise of zero mean and standard
C      deviation 'SIGMANOIS' may be added to the created object itself not to
C      the background.

```

C (2) It calculates 'ZMAX' the largest Z-coordinate or the largest
 C PICTUR(I,J) elements. Also it calculates IMIN, IMAX, JMIN AND JMAX,
 C integer values which are the I and J coordinates of the rectangle inside
 C which the object is enclosed.

C (3) It writes the obtained object matrix PICTUR(I,J) with or without
 C the added noise in a string form. It obtains the real vectors XPTS(K),
 C YPTS(K) and ZED(K), which are respectively the X, Y AND Z coordinates of
 C the nonzero values of matrix PICTUR(I,J).

C (4) It writes the obtained object matrix PICTUR(I,J) in an 'I*4'
 C format 'MAP' for display purposes under A given unit number 'IUNIT'.

C (5) It also writes A 'LOGICAL*1' file, for display purposes of the
 C object with or without the added noise. It is stored under the unit
 C number 'JUNIT'.

C-----

A2.2-6

C-----

C *****SUBROUTINE OBJECTROT does the following.

C (1) It inputs the vectors XPTS, TPTS and ZPTS for the nonzero
 C Z-coordinates of the object and rotates the coordinate points an angel
 C 'THETAROT' radians about the Z axis passing through the mean of the
 C X-coordinates and of the Y-coordinates of the object (XMEAN,YMEAN). The
 C output is matrix OBJECTROR(I,J).

C (2) It writes the obtained matrix PICTURROT(I,J) in a string form.
 C It obtains the real vectors XPTS(K), YPTS(K) and ZED(K), where are the X,
 C Y AND Z coordinates of the nonzero values of the object matrix
 C PICTURROT(I,J).

C (3) It calculates IMIN, IMAX, JMIN and JMAX integer values which are
 C the coordinates of the rectangle inside which the object is enclosed.

C (4) It writes the obtained object matrix PICTURROT(I,J) in an 'I*4'
 C format 'MAP' for display purposes under a given unit number 'IUNIT'.

C (5) It also writes for display a 'LOGICAL*1' file of the rotated
 C object with or without the added noise, under a unit number 'JUNIT'.

C-----

A2.2-7

C-----

C *****SUBROUTINE SMOOTHING does the following.

C This subroutine reads a real matrix pictur and outputs a real matrix
 C 'SPICTUR' which is the smoothed version of 'PICTUR'. Smoothing is done
 C by taking the average over a 3x3, 5x5 or 7x7 window. It also calculates
 C the largest element of the smoothed matrix.

C-----

A2.2-8

C-----
 C *****SUBROUTINE DISPLAY does the following.
 C This subroutine displays a given picture file such that each of its
 C elements is one of the characters '012345 6789 ' in an ascending order.
 C-----

A2.2-9

C-----
 C *****SUBROUTINE MOMENTAXS does the following.
 C This subroutine inputs a set of real nonzero Z-coordinates of the
 C object for each (X,Y). Then it obtains the X and Y object coordinates
 C 'XMEAN' and 'YMEAN', calculates the maximum and minimum moments of
 C inertia 'IMOMNTMAX', and 'IMOMNTMIN' at (XMEAN,YMEAN) and the angle
 C 'THETAMMT' which the axis of maximum moment of inertia makes with the
 C I-axis. Also it calculates the parameter 'SIGMAMMT'.
 C-----

A2.2-10

C-----
 C *****SUBROUTINE EQUATION does the following.
 C This subroutine calculates the coefficient matrix of an over-
 C determined system of linear equations. The least squares solution of
 C this equation gives the coefficients of the equation of the elliptic
 C (HYPERBOLIC) PARABOLOID which fits the nonzero Z-coordinates of the
 C object in the least squares sense.
 C The equation of the ELLIPTIC (HYPERBOLIC) PARABOLOID surface is
 C $Z=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y$ (A)
 C or
 C $Z=A(1)+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y$ (B)
 C-----

A2.2-11

C-----
 C *****SUBROUTINE NORMEQLS does the following.
 C Given is the real 'KPTS' by 'KOLX' matrix 'A' of rank KOLX.LE.KPTS
 C and a real 'KPTS' vector 'B' for the equation
 C $C*X = B.$

C This subroutine obtains the normal equation for $C^*X=B$ then calculates the least squares solution X of the overdetermined system $C^*X=B$.
 C For solving the normal equation, matrix $[C(TRANPOSE)*C]$ is factorized by
 C the CHOLESKY decomposition method.
 C-----

A2.2-12

C-----
 C *****SUBROUTINE ELLIPSE does the following.
 C Given is a general equation of an ELLIPSE (HYPERBOLA) this subroutine calculates MAJOR and the MINOR semi-axes of the conic and the angle, the MAJOR axis makes with the X-axis. The given equation of the ELLIPSE (HYPERBOLA) is
 C $0=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y$.
 C We wish to transform this equation to the standard form
 C $0=ANEW(1)+ANEW(4)*X*X+ANEW(6)*Y*Y$
 C from which the MAJOR and MINOR semi-axes and the ECCENTRICITY of the conic are obtained.
 C-----

A2.2-13

C-----
 C *****SUBROUTINE MOMENTKNT does the following.
 C This subroutine inputs a set of integer base contour points whose number is 'KONTPTS' and whose I and J coordinates are the integer arrays KONTXPTS and KONTYPTS. The contour forms a binary image. The contour points have the value 1 and the rest of the image zero.
 C The subroutine then calculates some invariant moment parameters of the base contour points and the parameter sigma_{knt}.
 C-----

A2.2-14

C-----
 C *****SUBROUTINE KONTOURS does the following.
 C This subroutine (1) finds out the outside and inside contours of a two dimensional image using a simple method. (2) It writes down the I and J-coordinates of the contour of the object in the form of a string of the I-coordinates and the corresponding string of the J-coordinates, and (3) the number of the obtained contour points.
 C-----

C The method of finding a contour point is as follows. The image
 C matrix is scanned column wise. Consider non-zero elements. Each of the
 C 8 neighbouring elements to this non-zero element as a center is examined.
 C If any of the 8 elements is zero, then the center element is a boundary
 C or a contour point.

C -----

A2.2-15

C -----
 C *****SUBROUTINE CONVOLS does the following.
 C Given is a 3-D object in the form of the image matrix 'SPICTUR'
 C whose (I,J) values are the Z-coordinates of the object.
 C This subroutine calculates (1) the 'SMOOTHED GAUSSIAN' 'SKEE(I,J)'
 C and (2) the 'SMOOTHED MEAN' 'SHEE(I,J)' curvatures at each point (I,J) of
 C the surface SPICTUR(I,J). The smoothed value is the average over a 7 by
 C 7 window.
 C It also calculates the 3 surfaces: (3) EDGMAP, (4) FLATMAP and (5)
 C ERRSURF which are respectively indications of the edges, the flatness and
 C a measure of the error in the least squares fit at (I,J).
 C
 C If desired by the user, this subroutine maps surfaces (1) SKEE, (2)
 C SHEE, (3) EDGMAP, (4) FLATMAP and (5) ERRSURF onto 'LOGICAL*1' files under
 C certain unit numbers (94, 95, ..., 98 respectively).
 C At each point (I,J) of 'SPICTUR', a quadratic surface of the form
 C $F(I,J) = A00 + A01*I + A01*J + A11*I*J + A20*(I^2-2) + A02*(J^2-2)$.
 C is fit in a least squares.
 C One calculates the partial derivatives of SPICTUR(I,J) W.R.T. U (I.E.
 C of I) and V (I.E. of J). These are denoted by FU, FV, FUU, FUV and FVV.
 C These in fact are obtained by convolving SPICTUR(U,V) with certain
 C weighted windows. SEE: P. R. BEAUDET, rotational invariant image
 C operators, 4TH ICPR, NOVEMBER 7-10, 1978, KYOTO, JAPAN, PP. 579-583.
 C -----

A2.2-16

C -----
 C *****SUBROUTINE SGNMAP does the following.
 C Given are the 'SMOOTHED gaussian curvature' 'SKEE' and the 'SMOOTHED
 C MEAN CURVATURE' 'SHEE' at each point (I,J), this subroutine calculates
 C the '8' combinations of the '+ 0 -' GAUSSIAN curvature and the '+ 0 -'

C mean curvatures and prints a map of string characters '12345678' for this
 C combination.

A2.2-17

C *****SUBROUTINE MOMENTS does the following.
 C This subroutine inputs a set of real Z-coordinates of the object
 C for each X and Y coordinates. It obtains the mean values of the X and Y
 C object coordinates 'XMEAN' and 'YMEAN' and calculates the 7 normalized
 C invariant moments AMMT(I), I=1:7.
 C REFERENCES:
 C (1) E. HALL, Computer image processing and recognition, academic press,
 C 1979, PP. 420-421.
 C (2) S. MAITRA, Moment invariants, proceedings of the IEEE, VOL-67, 1979,
 C PP. 697-699.



APPENDIX B. PARAMETERS DEFINITIONS

```

C-----
C
C NOTE: COMMON currently cannot be greater than 16382 words as SWPFCB reads
C       and writes it as a single record, and 16382 words is the biggest
C       allowed record length
C
C S500_COMMON_SIZE_I : Integer number = 16382 used to define the length of
C                     the FCB Integer Array.
C
C S500_COMMON_SIZE_R : Integer number = 8191 used to define the length of
C                     the RFCB.
C
C LUN : To use for indirect command files. This LUN will remain open over
C       several function calls!
C
C ERROR : An integer indicator (error code).
C
C         -2: Not available.
C         -1: Not acquired.
C         0: OK
C         1: Timeout
C         2: Invalid or non-responding device.
C         3: System dependent error;
C            error text returned in ERRSTR.
C-----
C
C BUFFER : An array of maximum length of S500_COMMON_SIZE_I ( 16382 ), the
C         biggest allowed record length. used to define the length of the
C         FCB.
C
C RBUFF : An array of maximum length of S500_COMMON_SIZE_R ( 8191 ), the
C         biggest allowed record length. used to define the length of the
C         RFCB.
C
C Ibuff : Matrix of [128 * 128] of type byte used to carry the data after
C         it has been read from the tape.
C
C Jbuff : A vector of length 1280 of type byte used to carry one line of
C         data at a time before we send it to the IIS screen.
C
C FILE1, FILE3-FILE10 : Each is a constant of type character used to
C                     indicates where MODEL'S data 1 to 9 are stored respectively.

```

C ex. FILE1 = 'FOR071'. FOR071 is the logical unit where the data
 C for MODEL number 1 are stored. FILE10 = 'FOR079'. FOR079 is the
 C logical unit where the data for MODEL number 9 are stored.
 C
 C FILE2 : Is a constant of type character used to indicate the fortran
 C logical unit FOR072, this is the place where the data will be
 C directed to after running the matching driver.
 C
 C FILENAME : Variable of type character used to indicate to one of
 C (FILE1-FILE10) at one time, FILENAME = FILE1, FILE3-FILE10 if
 C the user wants the MODEL data to be displayed. FILENAME = FILE2
 C if the user wants the object data to be displayed.
 C
 C NOTE: (for more explanation about the filename see the main menu of
 C Driver DRSHOW.FOR)
 C
 C ISW : An integer number represents the selected option from the main
 C menu. ISW = 1-6,99
 C
 C KOBJ : An integer number represents the desired object number for the
 C matching process. KOBJ = 1-42.
 C
 C NOISE = 0: No GAUSSIAN noise added to object matrix.
 C
 C SIGMANOIS = amount of standard deviation of added noise.
 C
 C ISMOOTH = 0: Input matrix is not smoothed
 C
 C NSMOOTH = size of smoothing window, 3x3,5x5 or 7x7.
 C
 C IFOR003 = 0 : File FOR003.DAT is not created.
 C
 C IDISPLAY = 0: The user does not wish to display the object using sub-
 C routine display.
 C
 C IMAPL = 0: MAPL matrix is not created.
 C
 C IMAPLC = 0: SKEE, SHEE, EDGMAP, FLATMAP & ERRSURF MAPS are not created.
 C
 C IKONT = 0: CONTOUR map is not printed.
 C
 C NORMSC = 0: The 7 calculated central moment parameters are not invariant
 C under a scale change.
 C
 C HEAD1, CHEADB : Are constants of type character used to make a grid and
 C print the output in a nice format.

C
 C IDIS : An integer number
 C 1- the user wants MODEL data to be displayed.
 C 2- the user wants tested OBJECT data to be displayed.
 C 3- the user wants to go to the MAIN MENU.
 C
 C IMODEL An integer
 C 1- the user wants the MODEL'S number and angle to be displayed
 C under the MODEL/OBJECT image.
 C 0- the user does not want the MODEL'S number and angle to be
 C displayed under the MODEL/OBJECT image.
 C
 C KS : An integer (1-8) represents the scale factor of the image buffer.
 C
 C IL : An integer represents the starting column on the IIS screen.
 C
 C JL : An integer represents the starting row on the IIS screen.
 C
 C SCALE : Array of length 9 of type character its first element represents
 C the number 000 the starting point of the scale axis, its second
 C element represents the first element + 32, its third element
 C represents the second element + 32, and so on.
 C
 C NOTE : The number 32 is the increment step size on the scale axis.
 C
 C PNUM : Array of length 13 of type character its first element represents
 C the number 01, the first parameter. Its second element
 C represents the number 02, the second parameter, and so on.
 C
 C X : A vector of length 256 its first element is the number 0, its
 C second element is the number 1, and so on.
 C
 C IPOS1 : An integer represents the starting column for the character font
 C in the Ibuff.
 C
 C IPOS2 : An integer represents the ending column for the character font in
 C the Ibuff.
 C
 C ICONST : An integer (0:255) represents the intensity of a particular
 C curve on the IIS screen.
 C
 C PENTE : A real number represents the step increment on the scale axis.
 C
 C IPENTE : An integer represents PENTE rounded to the nearest integer.
 C

C IROW An integer. It is the number of rows of the read or the created
 C (simulated) object or image.
 C
 C KOL An integer. it is the number of columns of read or created
 C (simulated) object or image.
 C
 C MAXROW An integer .GE.IROW.
 C
 C MAXKOL An integer .GE. KOL.
 C
 C MAXRAK An integer = $4 * (MAXROW + MAXKOL)$.
 C
 C MAXRK An integer = $MAXROW * MAXKOL$.
 C
 C IUNIT An integer. It is the unit number of the I*4 read or simulated
 C object or image.
 C
 C JUNIT An integer. It is the unit number of the LOGICAL*1 image of the
 C read image, for matrix 'MAPL'.
 C
 C NUM An integer, the number of the simulated object.
 C
 C NOISE An integer indicator specified by the user. Noise=0 indicates no
 C noise to be added to matrix pictur. NOISE.NE.0, indicates adding
 C GAUSSIAN noise with zero 'MEAN' and standard DEVIATION 'SIGMANOIS'
 C to pictur.
 C
 C SIGMANOIS A real parameter specified by the user. It is the standard
 C deviation of the added gaussian noise.
 C
 C PICTUR A real maxrow by maxkol matrix. its first irow rows and first KOL
 C columns contain the Z-coordinates of given object matrix 'PICTUR'
 C with or without the added noise.
 C
 C SPICTUR A real 'MAXROW' by 'MAXKOL' matrix which is a copy of pictur or
 C it is the smoothed pictur.
 C
 C PICTURROT A real maxrow by maxkol matrix. Its first irow rows and first
 C KOL columns are the Z-coordinates of the rotated object
 C 'PICTURROT'.
 C
 C THETAROT An angle in radians, with the X-AXIS specified by the user for
 C which 'OBJECT' is rotated about the Z-axis passing through the
 C C.G. of pictur.
 C

C NSMOOTH An integer parameter specified by the 3, 5, or 7. It is the size
 C of the smearing window.
 C
 C ZMAX A real number which is the largest Z-coordinate of the object.
 C That is the largest PICTUR(I,J) element.
 C
 C XMAX A real number which is the largest X-coordinate of the file.
 C
 C XMIN A real number which is the smallest X-coordinate of the file.
 C
 C YMAX A real number which is the largest Y-coordinate of the file.
 C
 C YMIN A real number which is the smallest Y-coordinate of the file.
 C
 C IMIN, IMAX, JMIN and JMAX
 C integer values which have the same meanings respectively as XMAX,
 C XMIN, YMAX and YMIN.
 C
 C XMEAN A real number which is the mean value of the X-coordinates of the
 C object surface.
 C
 C YMEAN A real number which is the mean value of the Y-coordinates of the
 C object surface.
 C
 C ZMEAN A real number which is the mean value of the Z-coordinates of the
 C object surface.
 C
 C VOLUM A real number which is the volume of the object.
 C
 C MAP An integer I*4 'MAXROW' by 'MAXKOL' matrix whose first irow rows
 C and first kol columns contain the output object matrix in non-
 C formatted format under the unit number iunit.
 C
 C MAPL A LOGICAL*1 'MAXROW' by 'MAXKOL' matrix whose first irow rows and
 C first kol columns contain the output object matrix in
 C non-formated format under the unit number junit for display
 C purposes.
 C
 C IMAPL An integer indicator specified by the user. IMAPL=0 indicates that
 C LOGICAL*1 matrix 'MAPL' is not to be created. If IMAPL.NE.0, it is
 C created.
 C
 C KPTS An integer. It is the number of nonzero values in the matrix of
 C the object.
 C

C XPTS A real vector of length maxrk. Its first KPTS elements are the
 C X-coordinates of the nonzero values of the object matrix
 C PICTUR(I,J).
 C
 C YPTS A real vector of length MAXRK. Its first KPTS elements are the
 C Y-coordinates of the nonzero values of the object matrix
 C PICTUR(I,J).
 C
 C ZED A real vector of length MAXRK. Its first KPTS elements are the
 C Z-coordinates of the nonzero values of the object matrix
 C PICTUR(I,J).
 C
 C KONTPTS An integer. It is the number of kontour points of the given
 C image.
 C
 C KONTXPTS An integer vector of length MAXRK. Its first kontpts elements
 C are the I-coordinates of the kontour points.
 C
 C KONTYPTS An integer vector of length MAXRK. Its first kontpts elements
 C are the J-coordinates of the kontour points which correspond to
 C the KONTXPTS points.
 C
 C KOLX An integer .EQ.NUMBER of columns in the coefficient matrix. For
 C equation (A) above, KOLX=6 and for equation (B) KOLX=4.
 C
 C MAXX An integer .GE. (KOLX+1). we take MAXX=7.
 C
 C C A real MAXRK by MAXX matrix. Its first 'KPTS' rows and first
 C 'KOLX' columns contain the coefficient matrix of the approximating
 C equation.
 C
 C B A real maxrk vector whose first KPTS contain vector 'B' of the
 C given equation $C \cdot X = B$.
 C
 C X A real 'MAXX' vector whose first KOLX elements are the least
 C squares solution of the system $A \cdot X = B$.
 C
 C RES A real MAXRK vector whose first KPTS elements are the residual
 C vector $(B - A \cdot X)$ of $A \cdot X = B$.
 C
 C RHO A real number which is the L2 NORM of the residual vector RES.
 C
 C R A real 'MAXX by MAXX' matrix whose first (KOLX+1) rows and first
 C 'KOLX' columns contain the product $A(\text{TRANPOSE}) \cdot (A|B)$.
 C

C Y A real MAXX working vector.
 C
 C EL A real 'MAXX by MAXX' matrix. Its first KOLX rows and first KOLX
 C columns contain the lower triangular matrix in the CHOLESKY
 C decomposition of the first KOLX columns of matrix R which is
 C $EL*EL(TRANSPOSE)$.
 C
 C A A real vector of length MAXX. Its first KOLX elements are the
 C coefficients of the equation of the ellipse (HYPERBOLA).
 C
 C ANEW A real vector of length MAXX. Its first KOLX elements are the
 C coefficients of the equation of a standard ellipse (HYPERBOLA).
 C That is in our case
 C $0 = ANEW(1) + ANEW(4)*X*X + ANEW(6)*Y*Y$
 C
 C ASEMIMAJ A real number, the SEMI MAJOR axis of the conic.
 C
 C BSEMIMIN A real number, the SEMI MINOR axis of the conic.
 C
 C THETAELL A real number. It is the angle (IN RADIANS) the MAJOR axis of
 C the ellipse (HYPERBOLA) makes with the X-axis.
 C
 C ECCENT The calculated ECCENTRICITY of the ellipse or the HYPERBOLA.
 C
 C MATCHAR A CHARACTER*1 MAXROW by MAXKOL matrix whose first IROW rows and
 C first KOL columns contain the output character matrix.
 C
 C NORMSC An integer specified by the user. If NORMSC=0, the 7 central
 C moment parameters are not invariant under a scale change. If
 C NORMSC.NE.0, they are invariant under a scale change.
 C
 C AMMT A real vector of length 7. Its elements are the 7 parameters
 C which are normalized invariant central moments for the given
 C object.
 C FU A real maxrow by maxkol matrix. Its first IROW rows and first KOL
 C columns contain the first partial derivative $DSPICUR(U,V)/DU$.
 C
 C FV A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the first partial derivative $DSPICUR(U,V)/DV$.
 C
 C FUV A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the second partial derivative $D2SPICUR(U,V)/DUDV$.
 C
 C FUU A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the second partial derivative $D2SPICUR(U,V)/DU2$.
 C

C FUV A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the second partial derivative D2SPICTUR(U,V)/DV2.
 C
 C EKEE A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the values of the GAUSSIAN curvature at elements
 C (I,J).
 C
 C EHEE A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the values of the SMOOTHED mean curvature at
 C elements (I,J).
 C
 C ABIGK MAX $\left| \text{SKEE}(I,J) \right|$
 C ABIGH MAX $\left| \text{SHEE}(I,J) \right|$
 C
 C AMAXK,AMAXH,AMAXEDG,AMAXFLAT,AMAXERRS :
 C MAX SKEE(I,J), MAX SHEE(I,J), MAX EDGMAP(I,J), MAX FLATMAP(I,J)
 C and MAX ERRSURF(I,J) respect.
 C
 C AMINK,AMINH,AMINEDG,AMINFLAT,AMINERRS :
 C MIN SKEE(I,J), MIN SHEE(I,J), MIN EDGMAP(I,J), MIN FLATMAP(I,J)
 C and MIN ERRSURF(I,J) respect.
 C
 C V A real working vector of dimension MAXROW.
 C
 C MATSGN A CHARACTER*1 MAXROW by MAXKOL matrix. Its first IROW rows and
 C first KOL columns contain the 8-LEVEL sign matrix values at
 C (X,Y).
 C
 C EDGMAP A real MAXROW by MAXKOL matrix. Its first IROW rows and first KOL
 C columns contain the value $\text{SQRT}(G) = \text{SQRT}(1+\text{FU}*\text{FU}+\text{FV}*\text{FV})$ at element
 C (I,J). This is A measure of an EDGE MAGNITUDE at (I,J).
 C
 C FLATMAP A real MAXROW by MAXKOL matrix. Its first IROW rows and first
 C KOL columns contain the value : $Q =$
 C $(\text{FUU}*\text{FUU}+2*\text{FUV}*\text{FUV}+\text{FVV}*\text{FVV})$ AT ELEMENT (I,J). This is a measure
 C of surface FLATNESS at (X,Y).
 C
 C ERRSURF A measure of the ERROR in the LEAST squares fit. = $\text{A00} - \text{FUJ} -$
 C $\text{FVV} - \text{SPICTUR}(I,J)$. It is the result of the convolution with the
 C 5 by 5 window
 C

1	-65	10	35	10	-65
---	10	85	110	85	10
875	35	110	-740	110	35
	10	85	110	85	10
	-65	10	35	10	-65

C
 C however, instead, we here calculate ERRSURF as the result of the
 C unsharp mask (SQUARES). SEE E.L HALL, Computer image processing
 C and recognition, academic PRESS, 1979, PP. 202-204.
 C
 C IMAPLC AN integer indicator specified by the user. If IMAPLC.NE.0, this
 C subroutine maps 'SURFACES' : (1) SKEE, (2) SHEE, (3) EDGMAP,
 C (4) FLATMAP and (5) ERRSURF onto 'LOGICAL*1' files under certain
 C unit numbers (94, 95,...98 respectively), if IMAPLC.EQ.0.
 C
 C NSIDE An integer, the side length of the square convolution window. We
 C take NSIDE=5.
 C
 C ZSIGMA A real number which is the standard deviation or the square root
 C of the variance of the Z coordinates of the object.
 C
 C SIGMAKNT A real number which is the standard deviation or the square
 C root of the variance of the X and Y coordinates of the contour of
 C the base silhouette of the object. See TEXT.
 C
 C MMTMAXKNT An integer, the maximum MOMENT of INERTIA of the contour of
 C the base silhouette about an axis through the centroid of the
 C contour.
 C
 C MMTMINKNT An integer, the minimum MOMENT of INERTIA of the contour of
 C the base silhouette about an axis through the centroid of the
 C contour.
 C
 C THETAKNT A real number. It is the angle (IN RADIANS) between the axis
 C of maximum MOMENT of INERTIA of the contour of the base silhouette
 C with the present I-axis.
 C
 C SIGMAMMT A real number which is the standard deviation or the square
 C root of the variance of the X and Y coordinates of the object
 C surface. See TEXT.
 C
 C IMOMNTMAX An integer, the maximum MOMENT of INERTIA of the object about
 C an axis through (XMEAN,YMEAN).
 C
 C IMOMNTMIN AN integer, the minimum MOMENT of INERTIA of the object about
 C an axis through (XMEAN,YMEAN).
 C
 C THETAMMT A real number. It is the angle (IN RADIANS) between the
 C axis of maximum MOMENT of INERTIA and the present I-axis.
 C

C EPS A specified tolerance such that a calculated number Z is
C considered zero if $|Z| < \text{EPS}$. For the VAX 11/780 computer, EPS is
C usually taken 1.E-4 and 1.E-11 for SINGLE and DOUBLE precision
C respectively.
C
C INDD An integer indicator. INDD=0 indicates normal exit INDD=-1
C indicates that KPTS.LT.KOLX, this subroutine is not designed for
C underdetermined systems, or, MAXX.LT.(KOLX+1), or
C MAXX.GT.MAXRK. INDD=-2 is for premature exit due to rank
C deficiency of matrix a.
C
C INDE AN integer indicator. INDE=0 indicates that the given equation
C is that of an ellipse. INDE=1 or INDE=2 indicates that the given
C equation is of a HYPERBOLA. INDE=-1 indicates that the given
C equation is for neither an ellipse nor for a HYPERBOLA and the
C subroutine exits prematurely.
C
C IERRORO An integer ERROR indicator.
C
C IERRORC An integer ERROR indicator.
C-----

APPENDIX C. SOURCE LISTING OF THE DRIVER DRSHOW.FOR

```

C                                PROGRAM DRSHOW.FOR
C
C    THIS PROGRAM IS A DRIVER FOR MATCHING '3-D' OBJECTS OBTAINED BY A
C    LAZER RANGE CAMERA, USING CERTAIN INVARIANT CHARACTERISTICS OF THE
C    OBJECT. AGAIN THIS DRIVER CALLS THE MAIN SUBROUTINE 'DR3DRANGE'.
C
C-----
C *** ALSO COMMAND FOR IMAGE DISPLAY ON THE IIS COLOUR MONITOR.
C
C *** DISPLAY OBJECT ON THE TERMINAL SCREEN, ADDED ON
C
C *** OPTION TO CREATE THE FOR003.DAT, ADDED ON
C
C-----
C    INCLUDE 'DUA0:[IIS.VMS]PARAMETER.INC'
C
C    Common area used in process communication
C
C    INTEGER FCB (S500_COMMON_SIZE_I)
C    REAL    RFCB(S500_COMMON_SIZE_R)
C    INTEGER MESS(15)
C    INTEGER LUN, ERROR
C    EQUIVALENCE (FCB, RFCB)
C    EXTERNAL S500_COND_HANDLER
C
C
C    INTEGER BUFFER(16382)
C    REAL    RBUFF(8191)
C    EQUIVALENCE (BUFFER, RBUFF)
C
C    INTEGER BUFSIZ, UNIT, QUIET
C    INTEGER ERRSTR(80)
C    INTEGER LEN
C
C    INTEGER PARMS(15)
C
C    INTEGER CHANLS(16)
C
C    Insert program declarations and data statements here
C
C    BYTE IBUFF(128,128)
C    BYTE JBUFF(1280)
C

```

```

CHARACTER*40 FILENAME,FILE1,FILE2,FILE3,FILE4,FILE5,FILE6,FILE7,
.FILE8

```

C

```

CHARACTER*40 FILE9,FILE10

```

C

```

DATA FILE1,FILE3,FILE4,FILE5,FILE6,FILE7,FILE8,FILE9,FILE10/'FOR071',
.'FOR072','FOR073','FOR074','FOR075','FOR076','FOR077','FOR078',
.'FOR079'/

```

C

```

DATA FILE2/'FOR070'/

```

C

```

LOGICAL*1 MAPL(128,128)
DIMENSION DUMPMAP(128,128)

```

C

```

DIMENSION PICTUR(128,128),SPICTUR(128,128),FU(128,1128),FV(128,128),
FUV(128,128),FUU(128,128),FVV(128,2128),V(128),EKEE(128,128),EHEE(128,
128),SKEE(128,3128),SHEE(128,128),EDGMAP(128,128),FLATMAP(128,128),
4ERRSURF(128,128),MAP(128,128)

```

C

```

DIMENSION C(16384,7),CK(1024,7),X(7),XNEW(7),R(7,7),IRES(16384),
RESK(1024),W(16384),DM(16384),EL(7,7),Y(7)2,AMMT(7),AMMTKNT(7)

```

C

```

DIMENSION XPTS(16384),YPTS(16384),ZED(16384)
DIMENSION PICTURROT(128,128),XROT(16384),YROT(16384)

```

C

```

DIMENSION ERRORR(9,22),SHOW(22),IS(9),IC(50),ISCALEDAT(15),
.SCALE(15)

```

C

```

DIMENSION DATAMODEL(9,15),IMATCH(15),IBEST(2,9)

```

C

```

INTEGER KONTXPTS(1024),KONTYPTS(1024),OBJECTSCL(15,9)

```

C

```

CHARACTER*1 MATSGN(128,128),MATCHAR(128,128),KONTMAP(128,128)

```

C

```

CHARACTER STRING*11,CPOST*3,CHEAD*74,CHEAD1*34,CHEADB*2,CHEADB1*76,
.CEHEAD*74

```

C

```

DATA STRING,CHEAD,CHEAD1,CHEADB,CHEADB1,CEHEAD/'0123456789 ',' +--
.-----+
.,-----, ', ' ', ' !
. !', ' +*****
*****+/
DATA DATAMODEL/39.9,30.6,31.5,29.7,32.6,36.3,39.7,.44.6,33.9,
10.56,.44,.75,.80,.81,.85,.89,.65,0.81,
249.5,35.2,39.5,39.3,40.9,48.7,42.5,41.3,36.9,

```

341.1,31.6,26.0,23.7,23.7,25.7,19.6,31.5,21.8,
 45990.,3255.,2862.,2653.,2780.,3710.,2456.,4222.,2200.,
 5100361.,38415.,32026.,38530.,39458.,63904.,63415.,149539.,51039.,
 616.73,11.8,11.19,14.52,14.19,17.22,25.82,35.42,23.20,
 711.76,4.60,5.38,7.5,6.26,5.84,8.28,5.18,8.85,
 840.16,32.48,30.69,29.93,31.62,35.69,29.48,36.47,28.23,
 926.16,24.03,25.19,18.79,21.31,25.87,23.26,25.28,18.84,
 138127888.,11175677.,15291277.,7628774.,13377435.,32141812.,28390340.,
 .60383436.,13001429.,
 230569152.,11004081.,5027503.,5971438.,4535967.,10614624.,5923321.,
 .35169552.,5111769.,
 3504963.,146194.,194177.,168145.,208524.,298850.,183276.,238124.,
 .132446.,
 4393837.,114630.,110644.,75887.,79049.,125851.,63064.,150414.,54175.,
 541.93,41.25,28.39,34.87,44.76,56.41,39.59,48.72,34.98/

C SCALED DATAMODEL MATRIX

DATA OBJECTSCL/

.228,160,255,255,255,171,120,255,255,255,161,222,255,
 .255,190,175,126,181,196,139,66,85,100,206,234,47,80,
 .74,74,186,180,215,203,161,122,55,81,117,195,246,65,36,
 .98,72,128,170,229,202,147,113,66,105,163,190,183,32,
 .43,85,49,158,186,232,211,147,118,67,102,136,201,208,
 .56,33,105,51,202,208,244,251,159,158,109,124,127,227,
 .252,136,77,151,81,255,227,255,219,122,105,108,186,
 .180,187,227,120,43,93,41,179,255,186,213,195,180,255,
 .255,112,232,246,255,255,120,97,220,194,232,190,135,
 .94,87,167,192,179,184,55,37,67,35,158/

C

INTEGER COL_START,ROW_START
 INTEGER COUNT,XINIT,YINIT,GRPSEL,CHMASK,BITMSK,ATM
 INTEGER XADV,YADV,ZADV,PMODE,RETRAC,XFOPT,KS
 INTEGER*2 X_SPLIT, Y_SPLIT
 INTEGER*2 QUADA(3), QUADB(3), QUADC(3), QUADD(3)

C-----

101 FORMAT(40H0 *****START DRIVER DRSHOW*****)
 102 FORMAT(24H IMATCH(I), I=1:15)
 103 FORMAT(24H IS(I) , I=1:9)
 104 FORMAT(1X,15I5)
 105 FORMAT(11H IUNIT =,I5)
 106 FORMAT(13H K IUNIT,/I5,I6)
 107 FORMAT(23H K IUNIT THETAROT,/I5,I6,4X,F8.4)
 108 FORMAT(46H IMAPL IMAPLC IKONT ISMOOTH NSMOOTH,/5(I7,2X))
 109 FORMAT(32H NORMSC NOISE SIGMANOIS ,/2X,I6,I7,2X,F8.2)
 110 FORMAT(T3,'!',25H IUNIT IS FOR OBJECT NO=,I5,T72,'!')
 111 FORMAT(T3,'!',40H MAJOR AXIS MAKES WITH I-AXIS ANGLE =
 .,F8.5,T72,'!')

```

112 FORMAT(T3,'!',46H MAJOR AXIS OF BASE MAKES WITH I-AXIS ANGLE =
    .,F8.5,T72,'!')
113 FORMAT(1X,8E9.2)
114 FORMAT(T3,'!',T5,F5.2,T9,F7.2,6F9.0,T72,'!')
115 FORMAT(22H EXEC. DRSHOW TIME =,F15.5,7HSECONDS)
116 FORMAT(22H EXEC. DRSHOW TIME1 =,F15.5,7HSECONDS)
117 FORMAT(T3,'!',T5,'RADS',T11,'ECCENT',T19,'ASEMIMAJ',T28,'BSEMIMIN',
    .T38,'KPTS',T46,'VOLUM',T58,'ZM',T66,'ZSGA',T72,'!')
118 FORMAT(T3,'!',T5,'SIGMAKNT',T15,'SIGMAMMT',T26,'MMTMAX',T36,'MMTMIN',
    .T46,'MMTMXKN',T55,'MMTMNKN',T66,'RADR',T72,'!')
119 FORMAT(T3,'!',2X,F7.2,T14,F7.2,T23,4F10.0,T66,F5.2,T72,'!')
120 FORMAT(30H AMMT(I), I=1:7 , KNTPTS )
121 FORMAT(T3,'!',T30,'OBJECT DATA',T72,'!')
122 FORMAT(T3,'!',T30,'MODEL DATA',T72,'!')
123 FORMAT(F3.1)
124 FORMAT(T3,'!',T30,'-----',T72,'!')
125 FORMAT(40H0 *****EXIT DRIVER DRSHOW***** )
126 FORMAT(T3,'!',T30,'-----',T72,'!')
127 FORMAT(' START OF FOR003.DAT FOR IUNIT =',I3)
128 FORMAT(1X,256A1)
129 FORMAT(T15,A34)
130 FORMAT(A72)
131 FORMAT(A2)
132 FORMAT(31H0 File to be printed is in --> ,A6,13H logical unit)
133 FORMAT(28H0 THE OBJECT IS ROTATED --> ,F8.4,7H radian)
134 FORMAT(I2)
135 FORMAT(I1)
136 FORMAT(40H THIS IS A STRANGE OBJECT )
137 FORMAT(40H * * * * * )

C
C-----C
C
C CALL LIB$ESTABLISH(S500_COND_HANDLER)
C
C-----< Initialize FCB >-----C
C Routine to initialize the FCB for the System 500 V3.
C
C CALL FCBSET (FCB)
C
C This routine is used to move a string of bytes from one array to another.
C The routine should be used to enter a Hollerith string into an array.
C
C CALL MOVCH (16HUNIT , BUFFER, 0, 0, 16)
C CALL PARM (FCB, PARMS, UNIT, BUFFER, 0, 1, 1, 1)

```

```

UNIT = 3
CALL DEFINE (FCB, PARMS, 1, 1)
C
C- Store Unit Number Information in Display portion of FCB.
C
FCB(190) = UNIT
C
C----- Initialize channel management data structures -----C
C
DO I = 1,16
FCB (129 + I) = -1
FCB (145 + I) = IXOR (32767, -1)
END DO
C
FCB (127) = 0
FCB (129) = 0
C
C----- Initialize Model 65 options -----C
C
CALL OPM75 (FCB, UNIT)
C
C Acquire and initialize display
C Display available?
C
CALL GTM70 (FCB, ERROR, ERRSTR, LEN)
FCB(126) = 0
C
C Routine to interpret display error codes and invoke the appropriate
C DOOM message. If the error code indicates success, this routine will
C simply return.
C
CALL M7OERR (FCB, ERROR, ERRSTR, LEN)
FCB(126) = 1
C
C---- Setup display for 1280 mode, red to mem 0,
C----- green to mem 1, blue to mem 2
C-----
C
MODE = 0 ! 0: 1280 MODE, 1: 640 MODE
C
CALL MOD640(FCB,MODE,0,0)
C
CHANLS(1) = 0
CHANLS(2) = 1
CHANLS(3) = 2

```

44

C

CALL SETUP(FCB,BUFFER,3,256,1,CHANLS,0,0)

C

CALL BCHAN(FCB,BUFFER,15,255)

CALL BCHAN(FCB,BUFFER,1,4095)

C

QUADA(1) = 0

QUADA(2) = 0

QUADA(3) = 0

C

QUADB(1) = 1

QUADB(2) = 1

QUADB(3) = 1

C

QUADC(1) = 0

QUADC(2) = 1

QUADC(3) = 2

C

QUADD(1) = 2

C

QUADD(2) = 2

QUADD(3) = 2

C

X_SPLIT = 1280

Y_SPLIT = 1024

C

C

This routine reads and writes the video split screen and video
select subunit.

C

CALL VIDSEL(FCB,BUFFER,X_SPLIT,Y_SPLIT,QUADA,
*QUADB,QUADC,QUADD,0,0)

C

C NOTE: The number of the invariant parameters (ICHARCTR) ranges from a
C minimum of 1 to a maximum of 20 parameters, at present time it is
C 15, but if you want to modify this number remember to change
C ICHARCTR below and the DATA segment in both routines (DRSHOW.FOR
C and DEMO.FOR)

NUMOBJECT = 42

MODELNUM = 9

ICHARCTR = 15

MAXX = 7

KOLX = 4

KOLXE = 4

KOLXS = 1

KOLXR = 2

IROW = 128

```

KOL      = 128
MAXROW   = 128
MAXKOL   = 128
MAXRAK   = 4*(MAXROW + MAXKOL)
MAXRK    = MAXROW*MAXKOL
PI       = 4.0*ATAN(1.0)
EPS      = 1.E-04
THETAZERO = 0.0
SCALEPAR = 255.0

```

C

```

DO J=1,ICHARCTR
  BIG = -1.0
  DO I=1,9
    VALU = DATAMODEL(I,J)
    IF(VALU.GT.BIG) BIG = VALU
  END DO
  SCALE(J) = SCALEPAR/BIG
END DO

```

C

C----- PROMPT THE USER TO SELECT FROM MENU -----

C

```

1 WRITE(3,2)
2 FORMAT(1H,'===== MENU ====='//
&' 1: Perform matching on selected objects data'/
&' 2: Display TESTED and/or MODEL image data'/
&' 3: Display TESTED curve data'/
&' 4: DEMO'/
&' 5: Clear CRT'/
&' 6: Funny colour'/
&' 7: Colour the IIS screen'/
&' 8: Save display'/
&' 9: Restore last saved display'/
&' 99: END '//
&'    ''Note at fresh start select either 1 or 4 '''/
&'    select menu number?', $)
READ(5,134,ERR=1) ISW
IF (ISW .EQ. 99) GO TO 1000
IF (ISW.LT.1.OR.ISW.GT.9) GO TO 1
IF (ISW.EQ.3) CALL DEMO (FCB,0,ISCALEDAT,ICHARCTR )
IF (ISW.EQ.4) CALL DEMO (FCB,1,ISCALEDAT,ICHARCTR )
IF (ISW.EQ.5) THEN
  CALL BCHAN(FCB,BUFFER,13,255) !Clear IIS DISPLAY
  CALL BCHAN(FCB,BUFFER,13,255) !Clear IIS DISPLAY
END IF
IF (ISW.EQ.6) CALL LINLUT( FCB, PI, IBUFF )

```

```

IF (ISW.EQ.7) CALL IISLINLUT ( FCB )
IF (ISW.EQ.8) CALL SAVERESTOR ( FCB, 0, 1, 2 )
IF (ISW.EQ.9) CALL SAVERESTOR ( FCB, 1, 0, 1 )
GO TO (3,22,1,1,1,1,1,1,1) ,ISW

```

C

C----- READ INPUT IMAGE FROM FILE -----

C

```

3 CONTINUE
4 PRINT*, ' Enter object number (1-42)   KOBJ > 42  exit'
  READ(5,134,ERR=4) KOBJ
  IF(KOBJ.GT.NUMOBJECT) GO TO 1
  IF(KOBJ.LT.1) GO TO 4

```

C

C

```

C MODEL 1 , KOBJ= 1:4      , IUNIT= 15:18
C MODEL 2 , KOBJ= 5:8      , IUNIT= 19:22
C MODEL 3 , KOBJ= 9:13     , IUNIT= 23:27
C MODEL 4 , KOBJ=14:18     , IUNIT= 28:32
C MODEL 5 , KOBJ=19:23     , IUNIT= 33:37
C MODEL 6 , KOBJ=24:28     , IUNIT= 38:42
C MODEL 7 , KOBJ=29:33     , IUNIT= 43:47
C MODEL 8 , KOBJ=34:37     , IUNIT= 51:54
C MODEL 9 , KOBJ=38:42     , IUNIT= 55:59

```

C

```

DO I=1,NUMOBJECT
  IF (I.LE.33)THEN
    IC(I) = I + 14
  ELSE
    IC(I) = I + 17
  END IF
END DO

```

C

```

IF (KOBJ.EQ.35) THEN
  THETAROT = PI/4.0
ELSE IF(KOBJ.EQ.36) THEN
  THETAROT = PI/2.0
ELSE IF(KOBJ.EQ.37) THEN
  THETAROT = 3.0*PI/4.0
ELSE IF(KOBJ.EQ.39) THEN
  THETAROT = PI/4.0
ELSE IF(KOBJ.EQ.40) THEN
  THETAROT = PI/2.0
ELSE IF(KOBJ.EQ.41) THEN
  THETAROT = 3.0*PI/4.0
ELSE IF(KOBJ.EQ.42) THEN
  THETAROT = PI

```

```

ELSE
    THETAROT = 0.0
END IF

C
    IF(KOBJ.GE.34.AND.KOBJ.LE.37) NUM = 2
    IF(KOBJ.GE.38.AND.KOBJ.LE.42) NUM = 6
C
    IUNIT = IC(KOBJ)
    JUNIT = 70
C NOISE = 0: NO GAUSSIAN NOISE ADDED TO OBJECT MATRIX.
    64 WRITE(3,65)
    65 FORMAT(' DO YOU WISH TO ADD NOISE TO THE PICTURE ?'/
    * '          1=YES   ;   0=NO ')
    READ (5,135,ERR=64) NOISE
    IF(NOISE.NE.1) GO TO 6
C SIGMAISOIS = AMOUNT OF STANDARD DEVIATION OF ADDED NOISE.
C
C----- ASK REQUIRED AMOUNT OF NOISE TO BE ADDED -----
C
    5 PRINT*, ' ENTER SIGMAISOIS (0.0 ,... , 4.0) ?'
    READ(5,123,ERR=5) SIGMAISOIS
    IF(SIGMAISOIS.LT.0.OR.SIGMAISOIS.GT.4)GO TO 5
C
C----- ASK IF SMOOTHING REQUIRED -----
C
C ISMOOTH = 0: INPUT MATRIX IS NOT SMOOTHED
C NSMOOTH = SIZE OF SMOOTHING WINDOW, 3X3,5X5 OR 7X7.
C
    6 WRITE(3,7)
    7 FORMAT(' DO YOU WISH TO SMOOTH THE PICTURE ?'/
    * '          1=YES   ;   0=NO ')
    READ (5,135,ERR=6) ISMOOTH
    SMAX = PMAX
    SMIN = PMIN
    IF(ISMOOTH.NE.1) GO TO 10
C
C----- ASK DESIRED WINDOW SIZE -----
C
    8 WRITE (3,9)
    9 FORMAT(' WHAT SIZE OF WINDOW DO YOU WANT ?'/
    * '          1- 3*3 WINDOW '/
    * '          2- 5*5 WINDOW '/
    * '          3- 7*7 WINDOW ')
    READ (5,135,ERR=8) NSM

```

```

      IF (NSM.EQ.1) NSMOOTH=3
      IF (NSM.EQ.2) NSMOOTH=5
      IF (NSM.EQ.3) NSMOOTH=7
      IF (NSM.LT.1.OR.NSM.GT.3) GO TO 8
C
C IFOR003 = 0 : --> FILE FOR003.DAT NOT CREATED.
  10 WRITE(3,11)
  11 FORMAT(' DO YOU WISH TO CREATE THE FOR003.DAT FILE ?'/
    * '          1=YES   ;   0=NO ')
    READ (5,135,ERR=10) IFOR003
C
C IDISPLAY = 0: THE USER DOES NOT WISH TO DISPLAY THE OBJECT USING
C      SUBROUTINE DISPLAY.
  12 WRITE(3,13)
  13 FORMAT(' DO YOU WISH TO DISPLAY THE OBJECT'/
    . '      ON THE TERMINAL SCREEN ?'/
    . '      1=YES   ;   0=NO ')
    READ (5,135,ERR=12) IDISPLAY
C IKONT = 0: CONTOUR MAP IS NOT PRINTED.
  62 WRITE(3,63)
  63 FORMAT(' DO YOU WISH TO DISPLAY THE BASE SILHOUETTE
    .CONTOUR'/
    . '      1=YES   ;   0=NO ')
    READ (5,135,ERR=62) IKONT
    IF(IFOR003.EQ.1)THEN
      WRITE(4,127) IUNIT
      WRITE(4,129) CHEAD1
      WRITE(4,131) CHEADB
      WRITE(4,131) CHEADB
    END IF
    IF(IUNIT.GT.47)THEN
      WRITE(3,107)KOBJ,IUNIT,THETAROT
    ELSE
      WRITE(3,106)KOBJ,IUNIT
    END IF
C
C IMAPL = 0: MAPL MATRIX IS NOT CREATED.
  IMAPL = 1
C IMAPLC = 0: SKEE, SHEE, EDGMAP, FLATMAP AND ERRSURF MAPS ARE NOT
C      CREATED.
  IMAPLC = 0
C NORMSC = 0: THE 7 CENTRAL MOMENT PARAMETERS ARE NOT INVARIANT UNDER A
C      SCALE CHANGE.
  NORMSC = 1
C
  IF(IFOR003.EQ.1)WRITE(4,108)IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH

```

```

IF(IFOR003.EQ.1) WRITE(4,109)NORMSC,NOISE,SIGMANOIS
WRITE(3,108)IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH
WRITE(3,109)NORMSC,NOISE,SIGMANOIS
ORIGIN = 0.0
TIME1 = SECNDS(0.0)    !initialize the clock to zero
CALL SETCLK(ORIGIN)    !initialize the clock to zero

```

C

C----- CALLING THE MAIN MATCHING SUBROUTINE -----

C

```

CALL      DR3DRANGE(IUNIT,JUNIT,NUM,IROW,KOL,EPS,
1MAXROW,MAXKOL,MAXRK,MAXRAK,MAXX,KOLX,DUMPMAP,PICTUR,
2SPICTUR,THETAROT,PICTURROT,KPTS,XPTS,YPTS,ZED,XROT,
3YROT,IKONT,KONTPTS,KONTXPTS,KONTYPTS,MMTMAXKNT,
4MMTMINKNT,THETAKNT,SIGMAKNT,FU,FV,FUV,FUU,FVV,V,EKEE,
5EHEE,SKEE,SHEE,EDGMAP,FLATMAP,ERRSURF,C,CK,X,XNEW,
6INDE,ECCENTK,ASEMIMAJ,BSEMIMIN,THETAELL,VOLUM,ZMEAN,
7INDK,ECCENTK,ASEMIMAJK,BSEMIMINK,THETAELLK,KOLXS,
8KOLXR,INDEE,ECCENTE,ASEMIMAJE,BSEMIMINE,THETAELLE,
9KOLXE,RADIUS, RADIUSR,ZSIGMA,IMOMNTMAX,IMOMNTMIN,
*THETAMMT,SIGMAMMT,NORMSC,AMMT,R,RES,RESK,W,DM,EL,Y,
1MAP,MAPL,IMAPL,IMAPLC,MATSGN,MATCHAR,KONTMAP,ISMOOTH,
2NSMOOTH,NOISE,SIGMANOIS,IDISPLAY)

```

C

```

CALL RDCLK(TIME,ORIGIN)
IF(IFOR003.EQ.1) WRITE(4,115)TIME
WRITE(3,115)TIME
DELTATM = SECNDS(TIME1)
WRITE(3,116)DELTATM

```

C

```

SHOW(1) = RADIUS
SHOW(2) = ECCENTK
SHOW(3) = ASEMIMAJK
SHOW(4) = BSEMIMINK
SHOW(5) = FLOAT(KPTS)
SHOW(6) = VOLUM
SHOW(7) = ZMEAN
SHOW(8) = ZSIGMA
SHOW(9) = SIGMAKNT
SHOW(10) = SIGMAMMT
SHOW(11) = FLOAT(IMOMNTMAX)
SHOW(12) = FLOAT(IMOMNTMIN)
SHOW(13) = FLOAT(MMTMAXKNT)
SHOW(14) = FLOAT(MMTMINKNT)
SHOW(15) = RADIUSR

```

C

50

```

DO 16 I=1,MODELNUM
  DO 15 K=1,ICHARCTR
    CON = (SHOW(K)-DATAMODEL(I,K))/DATAMODEL(I,K)
15  ERRORR(I,K) = ABS(CON)
16  CONTINUE
  DO 18 K=1,ICHARCTR
    SMALL = 1.E09
    DO 17 I=1,MODELNUM
      IF(ERRORR(I,K).GT.SMALL) GO TO 17
      SMALL = ERRORR(I,K)
      MATCH = I
17    CONTINUE
      IMATCH(K) = MATCH
18  CONTINUE
C
  DO 19 I=1,MODELNUM
19  IS(I) = 0
C
  DO 20 J=1,ICHARCTR
    I = IMATCH(J)
    IS(I) = IS(I) + 1
20  CONTINUE
  WRITE(3,102)
  WRITE(3,104)(IMATCH(I),I=1,ICHARCTR)
  WRITE(3,103)
  WRITE(3,104)(IS(I),I=1,MODELNUM)
C
  MAX = -1
  DO I=1,MODELNUM
    K = IS(I)
    IF (K.GE.MAX) MAX = K
  END DO
  MPAR = 3
  IF(MAX.LT.MPAR) THEN
    IF(IFOR003.EQ.1) THEN
      WRITE(4,137)
      WRITE(4,136)
      WRITE(4,137)
    END IF
    WRITE(3,137)
    WRITE(3,136)
    WRITE(3,137)
  END IF
C
  IF(IFOR003.EQ.1) THEN

```

```

WRITE(4,131) CHEADB
WRITE(4,102)
WRITE(4,104)(IMATCH(I),I=1,ICHARCTR)
WRITE(4,103)
WRITE(4,104)(IS(I),I=1,MODELNUM)
WRITE(4,131) CHEADB
END IF

C
C   Scale the results of SHOW vector to the right magnitude and store
C   them in ISCALEDAT vector, in order to prepare them for plotting in
C   graphics form (Display TESTED curve data).
C
DO I=1,ICHARCTR
  ISCALEDAT(I) = ININT(SCALE(I)*SHOW(I))
  IF (ISCALEDAT(I) .GT. 255) ISCALEDAT(I) = 255
END DO

C
PRINT*, " "
DO J=1,2
  DO I=1,MODELNUM
    IBEST(J,I)=0
  END DO
END DO
DO J=1,MODELNUM
  JMAX=-1
  DO I=J,MODELNUM
    K=IS(I)
    IF(K.GT.JMAX) THEN
      IBEST(1,I)=K
      IBEST(2,I)=I
    END IF
  END DO
END DO
DO J=1,MODELNUM
  DO I=J,MODELNUM
    IF(IBEST(1,I).GE.IBEST(1,J)) THEN
      TEMP1=IBEST(1,J)
      TEMP2=IBEST(2,J)
      IBEST(1,J)=IBEST(1,I)
      IBEST(2,J)=IBEST(2,I)
      IBEST(1,I)=TEMP1
      IBEST(2,I)=TEMP2
    END IF
  END DO
END DO
END DO

```

```

DO I=1,MODELNUM
  CPOST = ' th'
  IF(I.EQ.1) CPOST=' st'
  IF(I.EQ.2) CPOST=' nd'
  IF(I.EQ.3) CPOST=' rd'
C
C   We here write the results in a tabular form.
C
C Note: If we want to display every possible match, then the IF statement
C below should be:
C   IF(IBEST(1,I).NE.0) THEN
C
      IF(I.LE.3.AND.IBEST(1,I).NE.0) THEN
        PRINT*, ' +-----+
        .-----+
          WRITE(3,21) I,CPOST,IBEST(2,I),IBEST(1,I)
21  FORMAT(T3,'+',4H THE,I2,A3,28H BEST MATCH IS FOR
        .MODEL NO.          I3,' ',11H MATCHES IN,I3,11H
        .PARAMETERS,T72,'+')
        PRINT*, ' +-----+
        .-----+
          IF(I.EQ.1) THEN
            WRITE(3,121)
            WRITE(3,124)
            WRITE(3,117)
            WRITE(3,114)(SHOW(II), II=1,8)
            WRITE(3,118)
            WRITE(3,119)(SHOW(II), II=9,15)
          END IF
          WRITE(3,122)
          WRITE(3,126)
          WRITE(3,117)
          WRITE(3,114)(DATAMODEL(IBEST(2,I),II), II=1,8)
          WRITE(3,118)
          WRITE(3,119)(DATAMODEL(IBEST(2,I),II), II=9,15)
          PRINT*, ' I
          I
          IF (I.EQ.1)THEN
            WRITE(3,111)THETAMMT
            WRITE(3,112)THETAKNT
          END IF
          PRINT*, ' I
          I
          END IF
        END DO

```

```

      PRINT*, ' +.....
*.....+
      PRINT*, '
      PRINT*, '

```

```

C.....

```

```

C

```

```

C CHECK IF CREATION OF THE FOR003.DAT FILE WAS DESIRED, IF SO, GO AHEAD,
C CREATE IT.

```

```

      IF(IFOR003.NE.1) GO TO 1
      DO I=1,MODELNUM
        CPOST = ' th'
        IF(I.EQ.1) CPOST=' st'
        IF(I.EQ.2) CPOST=' nd'
        IF(I.EQ.3) CPOST=' rd'

```

```

C

```

```

C-----
C      Note: if we want to display say the first three best, then the IF
C statement below should:

```

```

C      IF(IBEST(1,I).NE.0) THEN
C      IF(I.LE.3.AND.IBEST(1,I).NE.0) THEN
        WRITE(4,130),CHEAD
        WRITE(4,21) I,CPOST,IBEST(2,I),IBEST(1,I)
        WRITE(4,130),CHEAD
        IF(I.EQ.1) THEN
          WRITE(4,121)
          WRITE(4,124)
          WRITE(4,117)
          WRITE(4,114)(SHOW(II), II=1,8)
          WRITE(4,118)
          WRITE(4,119)(SHOW(II), II=9,15)
        END IF
        WRITE(4,122)
        WRITE(4,126)
        WRITE(4,117)
        WRITE(4,114)(DATAMODEL(IBEST(2,I),II), II=1,8)
        WRITE(4,118)
        WRITE(4,119)(DATAMODEL(IBEST(2,I),II), II=9,15)
        WRITE(4,130) CHEADB1
        IF (I.EQ.1) THEN
          WRITE(4,111)THETAMMT
          WRITE(4,112)THETAKNT
        END IF
        WRITE(4,130) CHEADB1
      END IF
    END DO

```

```

WRITE(4,130) CEHEAD
WRITE(4,131)CHEADB
WRITE(4,131)CHEADB

```

```

C-----
C----- Here ends the matching process -----
C-----

```

```

GO TO 1          !Put up the MENU on the screen

```

```

C-----
C----- DISPLAY OUTPUT IMAGE ON THE IIS -----
C-----

```

```

22 WRITE(3,23)
23 FORMAT(' DO YOU WISH TO DISPLAY: '/
. ' 1-The MODEL data ?'/
. ' 2-The OBJECT data ?'/
. ' 3-Exit (menu) '//
. ' select menu number : ', $)
READ(5,135,ERR=22) IDIS
IF(IDIS.LT.1.OR.IDIS.GT.3) GO TO 22
GO TO (24,25,1) ,IDIS

```

```

C
24 IF      (IUNIT .GE. 15 .AND. IUNIT .LE. 18) THEN
      FILENAME=FILE1 !MODEL 1
ELSE IF (IUNIT .GE. 19 .AND. IUNIT .LE. 22) THEN
      FILENAME=FILE3 !MODEL 2
ELSE IF (IUNIT .GE. 23 .AND. IUNIT .LE. 27) THEN
      FILENAME=FILE4 !MODEL 3
ELSE IF (IUNIT .GE. 28 .AND. IUNIT .LE. 32) THEN
      FILENAME=FILE5 !MODEL 4
ELSE IF (IUNIT .GE. 33 .AND. IUNIT .LE. 37) THEN
      FILENAME=FILE6 !MODEL 5
ELSE IF (IUNIT .GE. 38 .AND. IUNIT .LE. 42) THEN
      FILENAME=FILE7 !MODEL 6
ELSE IF (IUNIT .GE. 43 .AND. IUNIT .LE. 47) THEN
      FILENAME=FILE8 !MODEL 7
ELSE IF (IUNIT .GE. 51 .AND. IUNIT .LE. 54) THEN
      FILENAME=FILE9 !MODEL 8
ELSE IF (IUNIT .GE. 55 .AND. IUNIT .LE. 59) THEN
      FILENAME=FILE10 !MODEL 9
END IF

```

```

C
GO TO 40

```

```

C
25 GO TO (30,31,31,31,32,31,31,31,33,31,31,31,31,
.      34,31,31,31,31,35,31,31,31,31,36,31,31,
.      31,31,37,31,31,31,31,38,31,31,31,39,31,
.      31,31,31), KOBJ

```

C
 C IF HERE, WE WANT TO DISPLAY THE OBJECT --> ASSIGN THE CORRECT LOGICAL
 C UNIT (file name) TO FILENAME.

C

```

30 FILENAME = FILE1 !PLACE OF MODEL NO. 1
   GO TO 40
31 FILENAME = FILE2   !PLACE OF WORKING OBJECTS
   GO TO 40           ! (1,2,3,4,5,6,7,8 OR 9).
32 FILENAME = FILE3   !PLACE OF WORKING OBJECT NO. 2.
   GO TO 40
33 FILENAME = FILE4   !PLACE OF WORKING OBJECT NO. 3.
   GO TO 40
34 FILENAME = FILE5   !PLACE OF WORKING OBJECT NO. 4.
   GO TO 40
35 FILENAME = FILE6   !PLACE OF WORKING OBJECT NO. 5.
   GO TO 40
36 FILENAME = FILE7   !PLACE OF WORKING OBJECT NO. 6.
   GO TO 40
37 FILENAME = FILE8   !PLACE OF WORKING OBJECT NO. 7
   GO TO 40
38 FILENAME = FILE9   !PLACE OF WORKING OBJECT NO. 8.
   GO TO 40
39 FILENAME = FILE10  !PLACE OF WORKING OBJECT NO. 9.
   GO TO 40

```

C

```

40 DO II=1,7
   DO JJ=1,7
     IBUFF(JJ,II)=0
   END DO
END DO
OPEN (UNIT=1, NAME=FILENAME, READONLY, FORM=
*'UNFORMATTED', STATUS='OLD',CARRIAGECONTROL='NONE')
DO IY=1,128
   READ(1) (IBUFF(IY,IX),IX=1,128)
END DO

```

C

C----- MAKE A FRAME FOR THE IMAGE -----

C

```

DO JJ=1,128
  DO II=1,128
    IF(((JJ.EQ.2.OR.JJ.EQ.127).AND.(II.GT.1.
.AND.II.LT.128)).OR. ((II.EQ.2.OR.II.EQ.127).
.AND.(JJ.GT.1.AND.JJ.LT.128))) IBUFF(JJ,II) = 255
  END DO
END DO

```

56

```

C      IF(IUNIT.GT.47) THEN
C      IF(IDIS.EQ.1) THEN
C          WRITE (3,133) THETAZERO
C      ELSE
C          WRITE (3,133) THETAROT
C      END IF
C      ELSE
C          WRITE (3,132) FILENAME
C      END IF
C
C----- ASK IF DISPLAYING OBJECT'S IS DESIRED.
26 WRITE(3,27)
27 FORMAT(' DO YOU WISH TO DISPLAY THE MODEL'S NUMBER' /
.      ' and ANGLE UNDER THE MODEL IMAGE?' /
.      '          1=YES ; 0=NO ')
      READ (5,135,ERR=26) IMODEL
      IF (IMODEL.EQ.1) CALL CDISPLAY(IBUFF,KOBJ,IDIS)
C
C----- OUTPUT TO IIS -----
C
41 WRITE (3,42)
42 FORMAT (' *** OUTPUT TO IIS ***' //
.      '* SCALING FACTOR [1..8]?')
      READ (5,135,ERR=41) KS
      IF (KS.LT.1.OR.KS.GT.8) GO TO 41
43 WRITE (3,44)
44 FORMAT(' STARTING COLUMN ON IIS [1..896]?')
      READ (5,*) I1
      IF (I1.LT.1.OR.I1.GT.896) GO TO 43
      COL_START = 1024 - (128*KS-1+I1)
      IF(COL_START.GE.0) GO TO 45
      PRINT*, 'CHECK SCALING FACTOR AND STARTING COLUMN'
      GO TO 43
45 WRITE (3,46)
46 FORMAT(' STARTING ROW ON IIS [1..896]?')
      READ (5,*) J1
      IF (J1.LT.1.OR.J1.GT.896) GO TO 45
      ROW_START = 1024 - (128*KS-1+J1)
      IF(ROW_START.GE.0) GO TO 47
      PRINT*, 'CHECK SCALING FACTOR AND STARTING ROW'
      GO TO 45
47 WRITE (3,48)
48 FORMAT(' ARE YOU SURE ? YES (Y) or NO (N)')
      READ (5,49) IANS
49 FORMAT(A1)
      IF (IANS.EQ.'N'.OR.IANS.EQ.'n') GO TO 41

```

```

C
C----- SCALE THE PICTURE -----
C
COUNT=128
XINIT=11
YINIT=J1
GRPSEL=0
CHMASK=1
BITMSK=-1
ATM=0
XADV=3
YADV=3
ZADV=3
PMODE=1
RETRAC=0
XFOPT=0
M=0
L=1
C
DO J=1,128                !128 rows
  DO I=1,128              !128 pixels/row
    DO K=1,KS              !KS=scaling factor
      JBUFF(L)=IBUFF(J,I) !JBUFF=output map vector
      L=L+1
    END DO
  END DO
C
END DO
C
DO K=1,KS                  !repeat output Map vector KS times
C
  CALL IMAGE(FCB,JBUFF(1),COUNT*KS,XINIT,YINIT+M,
*             GRPSEL,CHMASK,BITMSK,ATM,XADV,YADV,
*             ZADV,PMODE,RETRAC,XFOPT)
C
  M=M+1
END DO
C
L=1
END DO
C
CLOSE(1)
GO TO 22
C
1000 WRITE(3,125)
STOP ' normal exit for FORTRAN program'
END

```